

Phase 3.1

Quick Start Guide

Phase Quick Start Guide Copyright © 2009 Schrödinger, LLC. All rights reserved.

While care has been taken in the preparation of this publication, Schrödinger assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Canvas, CombiGlide, ConfGen, Epik, Glide, Impact, Jaguar, Liaison, LigPrep, Maestro, Phase, Prime, PrimeX, QikProp, QikFit, QikSim, QSite, SiteMap, Strike, and WaterMap are trademarks of Schrödinger, LLC. Schrödinger and MacroModel are registered trademarks of Schrödinger, LLC. MCPRO is a trademark of William L. Jorgensen. Desmond is a trademark of D. E. Shaw Research. Desmond is used with the permission of D. E. Shaw Research. All rights reserved. This publication may contain the trademarks of other companies.

Schrödinger software includes software and libraries provided by third parties. For details of the copyrights, and terms and conditions associated with such included third party software, see the Legal Notices for Third-Party Software in your product installation at `$SCHRODINGER/docs/html/third_party_legal.html` (Linux OS) or `%SCHRODINGER%\docs\html\third_party_legal.html` (Windows OS).

This publication may refer to other third party software not included in or with Schrödinger software ("such other third party software"), and provide links to third party Web sites ("linked sites"). References to such other third party software or linked sites do not constitute an endorsement by Schrödinger, LLC. Use of such other third party software and linked sites may be subject to third party license agreements and fees. Schrödinger, LLC and its affiliates have no responsibility or liability, directly or indirectly, for such other third party software and linked sites, or for damage resulting from the use thereof. Any warranties that we make regarding Schrödinger products and services do not apply to such other third party software or linked sites, or to the interaction between, or interoperability of, Schrödinger products and services and such other third party software.

June 2009

Contents

Document Conventions	vii
Chapter 1: Introduction	1
1.1 About Phase	1
1.2 About this Document	1
1.3 Preparing for the Exercises	2
Chapter 2: Building a Pharmacophore Model	5
2.1 Starting the Exercises	5
2.2 Adding Ligands	6
2.3 Choosing the Active and Inactive Sets	8
2.4 Proceeding to Create Sites	9
2.5 Examining Feature Mappings	11
2.6 Examining and Modifying Feature Definitions	12
2.7 Adding New Feature Definitions	15
2.8 Adding New Feature Types	17
2.9 Creating Pharmacophore Sites	18
2.10 Proceeding to Find Common Pharmacophores	18
2.11 Changing the Number of Sites in Common Pharmacophores	20
2.12 Changing the Allowed Feature Frequencies	21
2.13 Examining Options for Finding Common Pharmacophores	22
2.14 Finding Common Pharmacophores	24
2.15 Proceeding to Score Hypotheses	24
2.16 Scoring Hypotheses	24
2.17 Viewing Hypotheses and Ligand Alignments	29
2.18 Proceeding to Build QSAR Model	32
2.19 Assigning Training and Test Set Memberships	33

2.20	Setting QSAR Model Options.....	34
2.21	Building the QSAR Models.....	35
2.22	Visualizing the QSAR Model	38
Chapter 3: Creating a 3D Database.....		43
3.1	Creating a New 3D Database.....	43
3.2	Adding Structures to the Database.....	43
Chapter 4: Finding Matches to a Hypothesis.....		47
4.1	Preparing for the Exercises.....	47
4.2	Choosing the Database and Hypothesis	48
4.3	Performing a Standard Search.....	50
4.4	Searching Among Existing Matches.....	50
4.5	Searching with Site and Conformer Creation	51
Chapter 5: Developing Pharmacophore Models from the Command Line.....		53
5.1	Pharmacophore Model Utilities	53
5.2	Creating a New Command-Line Project.....	55
5.3	The Master Data File	56
5.4	Defining Active and Inactive Sets	58
5.5	Defining QSAR Training and Test Sets.....	59
5.6	Creating Pharmacophore Sites.....	60
5.7	Finding Common Pharmacophores.....	62
5.8	Scoring Hypotheses with Respect to Actives.....	65
5.9	Scoring Hypotheses with Respect to Inactives.....	68
5.10	Clustering Hypotheses by Geometric Similarity	69
5.11	Building 3D QSAR Models.....	75

5.12 Visualizing QSAR Models	79
5.13 Applying Models to New Molecules	81
5.13.1 Standard Search	81
5.13.2 Flexible Search	84
5.14 Creating Excluded Volumes	85
5.14.1 Creating a Shell Around a Ligand	86
5.14.2 Using Inactives to Define Steric Clash Regions.....	87
5.14.3 Creating Excluded Volumes from a Receptor	89
 Chapter 6: Creating and Searching 3D Databases from the Command Line.....	 91
6.1 Database Utilities	91
6.2 Creating a New 3D Database.....	92
6.3 Adding Molecules to an Existing Database.....	95
6.4 Creating Conformers and Pharmacophore Sites	96
6.5 Deleting Molecules from a Database	97
6.6 Database Searching: Background.....	98
6.7 Running a find+fetch Database Search.....	100
6.8 Running a fetch Database Search	103
6.9 Creating a Flexible Database	105
6.10 Running a flex Database Search	106
6.11 Using Site Masks.....	109
6.12 Using Feature-Matching Rules	110
6.13 Using Feature-Matching Tolerances	112
6.14 Using Hypothesis-Specific Matching Tolerances	113
6.15 Working with Database Subsets.....	114
6.15.1 Restricting a Database Search to a Subset of Hits.....	115
6.15.2 Creating a Subset from Other Subsets with Logical Operations.....	116

6.15.3 Restricting Conformer and Site Creation Using Subsets 116

6.16 Working with Database Properties 118

Getting Help 121

Glossary..... 125

Document Conventions

In addition to the use of italics for names of documents, the font conventions that are used in this document are summarized in the table below.

Font	Example	Use
Sans serif	Project Table	Names of GUI features, such as panels, menus, menu items, buttons, and labels
Monospace	<code>\$SCHRODINGER/maestro</code>	File names, directory names, commands, environment variables, and screen output
Italic	<i>filename</i>	Text that the user must replace with a value
Sans serif uppercase	CTRL+H	Keyboard keys

Links to other locations in the current document or to other PDF documents are colored like this: [Document Conventions](#).

In descriptions of command syntax, the following UNIX conventions are used: braces { } enclose a choice of required items, square brackets [] enclose optional items, and the bar symbol | separates items in a list from which one item must be chosen. Lines of command syntax that wrap should be interpreted as a single command.

File name, path, and environment variable syntax is generally given with the UNIX conventions. To obtain the Windows conventions, replace the forward slash / with the backslash \ in path or directory names, and replace the \$ at the beginning of an environment variable with a % at each end. For example, `$SCHRODINGER/maestro` becomes `%SCHRODINGER%\maestro`.

In this document, to *type* text means to type the required text in the specified location, and to *enter* text means to type the required text, then press the ENTER key.

References to literature sources are given in square brackets, like this: [10].

Introduction

1.1 About Phase

Phase is a versatile product for pharmacophore perception, structure alignment, activity prediction, and 3D database searching. Given a set of molecules with high affinity for a particular protein target, Phase utilizes fine-grained conformational sampling and a range of scoring techniques to identify common pharmacophore hypotheses, which convey characteristics of 3D chemical structures that are purported to be critical for binding. Each hypothesis is accompanied by a set of aligned conformations that suggest the relative manner in which the molecules are likely to bind.

A given hypothesis may be combined with known activity data to create 3D QSAR models that identify overall aspects of molecular structure that govern activity. These models may be used in conjunction with the hypothesis to mine a 3D database for molecules that are most likely to exhibit strong activity toward the target.

Phase provides support for lead discovery, SAR development, lead optimization and lead expansion. Phase may also be used as a source of molecular alignments for third-party 3D QSAR programs.

Phase is integrated into Maestro, the graphical user interface (GUI) for all Schrödinger products. An overview of the general capabilities of Maestro is given in the [Maestro Overview](#). For more detailed information on Maestro, see the Maestro online help, the [Maestro User Manual](#), or the [Maestro Tutorial](#). For detailed information on Phase, see the [Phase User Manual](#).

1.2 About this Document

This document provides tutorial instruction in the three main Phase workflows, both from the Maestro GUI and from the command line. The first three chapters give instruction in using the GUI; the following two chapters give instruction in using the command-line tools.

- [Chapter 2](#) contains exercises on developing a pharmacophore model and building QSAR models.
- [Chapter 3](#) contains exercises on preparing a database for searching. The database will contain both single structures, without conformers and sites, and structures for which the conformers and sites are generated and stored in the database.

- [Chapter 4](#) contains exercises on searching the database for matches to a hypothesis, using a database and a hypothesis supplied with the distribution. You do not need to complete the exercises in [Chapter 2](#) or [Chapter 3](#) to work through this chapter.
- [Chapter 5](#) contains exercises on developing a pharmacophore model and associated QSAR models from the command line.
- [Chapter 6](#) contains exercises on creating and managing a 3D database, and searching that database for matches.

The command-line tutorials assume that you are already familiar with Phase concepts, whereas the GUI tutorials make fewer assumptions. Before running the command-line tutorials, you are encouraged to work through the GUI tutorials. However, a more self-contained set of command-line tutorials is available with the product distribution. Each tutorial consists of a PDF file containing the instructions and a gzipped archive (.tar) file containing the necessary files. The available tutorials are listed in [Table 1.1](#).

Table 1.1. List of command-line tutorials

Tutorial	Documentation	Archive file
Pharmacophore Model Development	pharm_tutorial.pdf	pharm_tutorial.tar.gz
Database Management and Searching	db_tutorial.pdf	db_tutorial.tar.gz

These tutorials can only be run on UNIX platforms.

1.3 Preparing for the Exercises

To do the exercises, you must have access to an installed version of Maestro 9.0 and Phase 3.1. For installation instructions, see the [Installation Guide](#). Before you start the tutorial, you must create a working directory, and copy files from the Phase distribution into this directory.

UNIX:

1. Set the SCHRODINGER environment variable to the directory in which Maestro and Phase are installed:

```
csh/tcsh:    setenv SCHRODINGER installation_path
sh/bash/ksh: export SCHRODINGER=installation_path
```

2. Change to a directory in which you have write permission.

3. Uncompress and extract the archive file for the tutorial into this directory.

```
tar -xzvf $SCHRODINGER/phase-vnnnnn/tutorial/phase_tutorial.tar.gz
```

Windows:

1. Open the folder in which you want to create the folder that serves as your working directory.

The default working directory used by Maestro is your user profile, which is usually set to C:\Documents and Settings*username* on XP and C:\Users*username* on Vista. To open this folder, do the following:

- a. From the **Start** menu, choose **Run**.
 - b. Enter %USERPROFILE% in the **Open** text box and click **OK**.
2. Under **File and Folder Tasks**, click **Make a new folder**.
You can also choose **File > Folder > New**.
 3. Enter a name for the folder.
If you want to create a folder inside this folder, open the folder and repeat steps 2 and 3.
 4. Open the folder that contains the tutorial files. This folder is in the Schrödinger software installation, which by default is installed at C:\Schrodinger2009.
 - a. Open an explorer window.
 - b. Navigate to the Schrödinger software installation.
 - c. Open the phase-v*version* folder (*version* is the 5-digit Impact version number), then open the tutorial folder inside that folder.
 5. Drag the phase_tutorial.tar.gz archive to the folder you created in [Step 3](#).

You can close the tutorial folder in this software installation now.

6. Extract the phase_tutorial.tar.gz archive into your working folder.

If you don't have a utility to perform this task, you might have to install one, such as WinZip or 7-zip.

Building a Pharmacophore Model

This chapter is designed to help you become familiar with the Develop Pharmacophore Model workflow of Phase. This workflow involves the identification of common pharmacophore hypotheses from a set of active ligands. A common pharmacophore hypothesis is a spatial arrangement of chemical features common to two or more active ligands, which is proposed to explain the key interactions involved in ligand binding. Each hypothesis identified by Phase is scored according to how well the active ligands superimpose when they are aligned on the features associated with that hypothesis. A high-scoring hypothesis might be used to search a 3D database for new potentially active molecules, or it might be used to align a series of ligands in order to create a 3D QSAR model.

The structures and data used in this portion of the tutorial were taken from *J. Med. Chem.* **2003**, 46, 716-726. The data set consists of 50 angiotensin AT₁ antagonists, divided into training and test sets of 25 ligands apiece. For convenience and clarity, ligand names have been assigned to indicate membership in the training set (train01, train02, ...) or the test set (test01, test02, ...). You will be developing pharmacophore and QSAR models from the training set, and applying them to the test set.

2.1 Starting the Exercises

To begin the tutorial, start Maestro and save the scratch project to preserve the results:

1. Start Maestro.

On Windows, double-click the Maestro icon. On Linux, enter the command:

```
$SCHRODINGER/maestro &
```

The Maestro main window is displayed.

2. Choose **Maestro > Change Directory** and navigate to the `phase_tutorial/at1` directory in the directory you created in [Section 1.3 on page 2](#).
3. Click the **Save As** toolbar button.



The Save Project As project selector opens.

4. In the File name text box, type `at1_tutorial`.
5. Click Save.

Maestro provides a wizard to guide you through the steps of the Develop Pharmacophore Model workflow, in the appropriate order. This workflow allows you to identify common pharmacophore hypotheses and create 3D QSAR models.

6. Choose Applications > Phase > Develop Pharmacophore Model from main window.

The Develop Pharmacophore Model panel is displayed.

At the bottom of the panel is a series of buttons labeled Prepare Ligands, Create Sites, and so on. These are the various steps in the Develop Pharmacophore Model workflow, and clicking on any enabled button takes you directly to that step. Currently, all buttons except Prepare Ligands are disabled. A button is enabled only if the prerequisite steps in the workflow have been successfully completed. The Step menu at the top of the panel can also be used to move around the workflow.

2.2 Adding Ligands

No structures have yet been added to the Phase workflow, so there are no entries in the Ligands table. The structures that you copied from the distribution must now be added.

1. Click the From File button, which is located near the top of the panel.

The Add From File file chooser is displayed.

2. Select the file `at1.maegz`, and click Open.

This file contains 50 ligands. These ligands have been prepared and conformers generated for each. If you are starting with 2D single structures, you would have to clean the structures and then generate conformers.

When you click add, the Choose Activity Property dialog box opens. If you intend to build QSAR models (which you will do in this tutorial), this is where you should select the property to use as the experimental activity variable.

3. Select the pIC50-Exp property.
4. Ensure that Convert property values is *not* selected.
5. Click OK.

The Choose Activity Property dialog box closes and the ligands are imported. This process takes a few minutes, and a progress dialog box is displayed during the import.

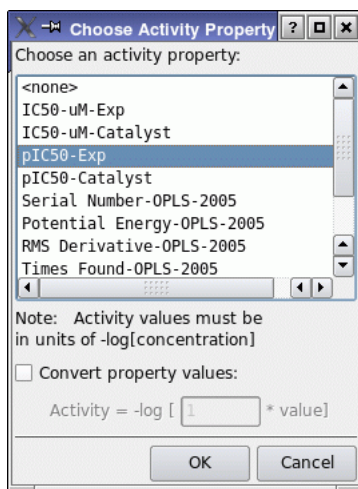


Figure 2.1. The Choose Activity Property dialog box.

The Ligands table is now filled with *copies* of the structures from the file. The Ligands table contains an *ln* column, which allows you to view one or more ligands in the Maestro Workspace. The Name column holds the same information as the Title column of the Project Table, and the Activity column contains the pIC50-Exp data.

The Pharm Set column indicates whether a molecule is in the set of actives used to identify common pharmacophore hypotheses, or in the set of inactives used to eliminate nondiscriminatory hypotheses, or in neither set. If ligands of widely varying activity are present, you would normally want to use only the most active ones in the set of actives. The most active ligands are assumed to contain the strongest binding, most important, or greatest number of pharmacophore features that are involved in binding to the protein target. The set of actives should contain as much structural diversity as possible, so that the resulting pharmacophore models are applicable across different chemical families.

The # Conformations column indicates how many conformations are present for each ligand. In this case, you imported multiple conformers for each ligand. When adding from the Project Table, you would normally see only a single conformation for each ligand unless you were selecting conformer sets.

The original set of 50 ligands contained a pair of duplicates: test17 and test24. The latter is missing from the Ligands table. What has happened here is that test17 and test24 were found to be chemically identical and were therefore merged into a single entry when the conformers were generated. It turns out that these two structures are also identical in the publication from which they were taken, so you will proceed with a reduced data set of 49 ligands and the knowledge that the experimental activity value for test17 might not be reliable.

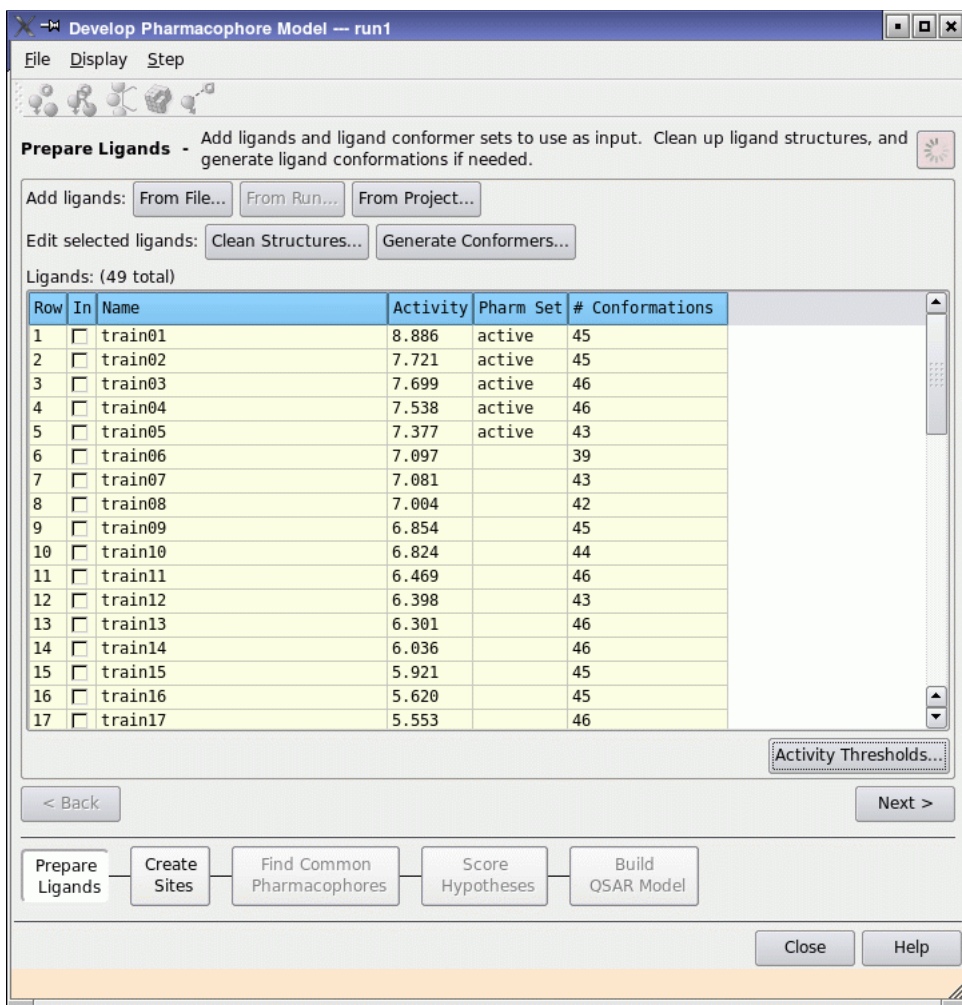


Figure 2.2. Ligands table in the Prepare Ligands step.

2.3 Choosing the Active and Inactive Sets

Only the most active compounds are normally considered when developing common pharmacophore hypotheses. Inactives can be used to eliminate hypotheses that do not provide a good explanation of activity on the basis of the pharmacophores alone. The active set determines the pool of pharmacophore models that are generated, and the initial scores that are assigned to them. The inactive set may be used subsequently to assign adjusted scores that reflect the degree to which the models distinguish actives from inactives. This is particularly useful if

everything in the active set is built on a common scaffold, which can give rise to a number of spurious pharmacophore models that have nothing to do with ligand binding.

In this section, you will set a threshold for actives of $IC_{50} \leq 50$ nM, which translates to $pIC_{50} \geq 7.3$, and a threshold for inactives of $pIC_{50} \leq 5.0$.

1. In the Develop Pharmacophore Model panel, click Activity Thresholds.

The Activity Thresholds dialog box appears, which allows you to specify the activity thresholds.

2. In the Active if activity above text box, type 7.3.
3. In the Inactive if activity below text box, type 5.0.
4. Click OK.

The Pharm Set column now has active for each ligand whose pIC_{50} value is greater than 7.3, and inactive for each ligand whose pIC_{50} value is less than 5.0. The column is blank for ligands whose activity falls between these values.

5. Clear the Pharm Set column for any test set ligands (test1 through test25) that are assigned to the active or the inactive set.

You can do this by selecting the assigned ligands and control-clicking in the column for one of ligands until the selected rows are blank in this column. The values cycle through active, inactive and blank when you click.

The test set ligands should not be part of the pharm set because the pharm set is used to develop the pharmacophore model, and the test set is used to validate the model.

Five ligands should now be in the active set: train01 through train05, and six ligands in the inactive set: train20 through train25.

2.4 Proceeding to Create Sites

Because the ligands you added to the run were multiconformer sets that had already been cleaned, there is no need to clean the structures or generate conformers. The next task is to identify and store the pharmacophore sites for each conformation.

- Click Create Sites in the Guide, or click Next.

There may be a delay of a few seconds before the panel is updated to the next step, while Phase does a fast mapping of pharmacophore features to the first conformation of each ligand. This mapping allows you to preview features before generating them for all conformations.

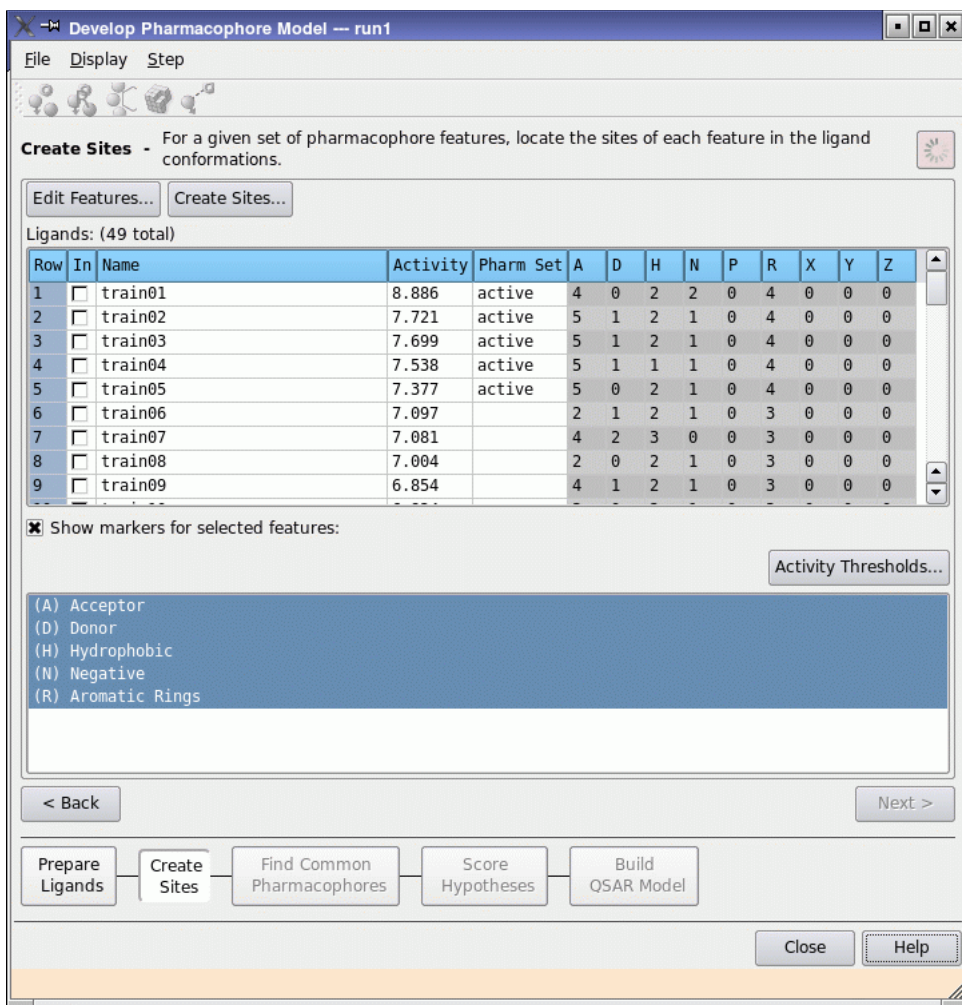


Figure 2.3. The Create Sites step.

Once the panel is updated, the Ligands table contains columns (A, D, N, etc.) showing the counts of each type of pharmacophore feature in each ligand. By default, Show markers for selected features is selected, and all features in the list below it are selected.

2.5 Examining Feature Mappings

To fully understand how pharmacophore models are developed, it is important to understand how the various pharmacophore features are mapped onto the ligand structures.

Note: This exercise is optional, and does not affect the final pharmacophore model.

1. If the Workspace is not empty, click the Clear Workspace button on the main toolbar.



2. Select (A) Acceptor in the features list under Show markers for selected features.
3. Place the third ligand (train03) in the Workspace by clicking the In check box in the third row of the Ligands table.

You should see various pink transparent spheres with arrows protruding from them. The spheres are centered on the hydrogen bond acceptor features of the ligand, and the arrows indicate the axes along which ideal hydrogen bonds would be formed.

4. Select (D) Donor in the features list to display hydrogen-bond donors.

A blue sphere appears on the hydroxyl hydrogen of this ligand. Once again, the arrow indicates the direction of the ideal hydrogen bond.

5. Select (H) Hydrophobic in the features list to display hydrophobic features.

A green sphere appears in the middle of the n-butyl chain attached to the imidazole ring, and another appears on the iodine atom. Since there is no directionality to the hydrophobic feature, it is represented as a sphere without an arrow.

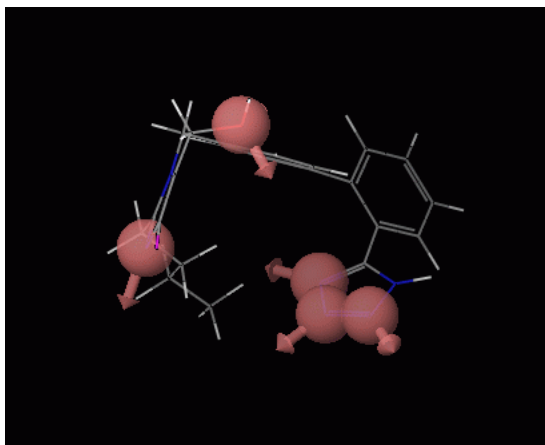


Figure 2.4. Acceptor features for ligand train03.

6. Select (N) Negative in the features list to display negatively-charged features.

A red sphere appears in the center of the tetrazole ring. While there is no ionic charge in this ligand structure, tetrazole is known to be acidic, and so it is perceived as a negative ionizable feature, much like a carboxylic acid.

7. Select (R) Aromatic Rings in the features list.

Orange toroids illustrate the locations of aromatic rings.

8. Select (A) Acceptor in the features list again.

2.6 Examining and Modifying Feature Definitions

The pharmacophore features that were visualized in the previous section are mapped to the structures using a set of topological feature definitions, which you will examine here.

1. Click Edit Features.

The Edit Features dialog box is displayed. This dialog box contains SMARTS patterns and other information that control the application of pharmacophore feature definitions. By default, definitions for the Acceptor (A) feature type are displayed; other feature types may be viewed by selecting from the Feature option menu. These are the built-in definitions provided with every installation of Phase, and they cannot be modified or deleted. They can, however, be *ignored*, as you will see.

Definitions near the top of the table have higher precedence than those closer to the bottom of the table. So, for example, if the first pattern maps a particular nitrogen in the ligand as an acceptor, that same nitrogen will not be mapped as an acceptor by any subsequent pattern. The vertical positions of the built-in features may not be changed, but user-defined patterns (discussed in the next section) may be moved using the arrowhead buttons below the Pattern list table.

The order of precedence does not apply to *excluded* patterns, i.e., patterns for which a check mark appears in the Exclude column. Excluded patterns, which will be discussed shortly, are processed before all other patterns.

To understand how the definitions are applied, consider the seventh entry in the table, for which the SMARTS pattern is [n;X2]([a])([a]). This 3-atom pattern is matched by an aromatic nitrogen and the two aromatic atoms to which it bonds.

2. Click the Mark box in the seventh row (the one with the pattern [n;X2]([a])([a])).

The atoms and bonds in the matching substructures are now marked in pink on the Work-space ligand (train03). The pink acceptor spheres, which should also be visible, clarify that the pattern is matched three times in the tetrazole ring.

The Geometry for this feature definition is **vector**, meaning that the feature is located on a single atom and has one or more directions associated with it—the directions of the possible hydrogen bonds. Hydrogen bond acceptors and donors are vector features, and aromatic rings are also vector features because the orientation of the ring is important. The other available geometries are **point** and **group**. These have no directionality and the associated features are located on either a single atom (point) or at the centroid of a group of atoms (group). Atom Numbers are referenced to the SMARTS pattern and are used in conjunction with Geometry to define the location of the feature. Projected Point Type designates the configuration of idealized hydrogen bonds for determination of vector orientations in acceptor and donor features.

Some feature definitions are checked as **Exclude**. Exclusion performs a logical NOT operation for matches to these patterns. To clarify what this means, the following steps show how **Exclude** definitions are applied.

3. Clear the **Mark** box for the previous SMARTS pattern `[n;X2]([a])([a])`.

The substructure markers disappear from the Workspace ligand.

4. Remove `train03` from the Workspace and replace it with `train01` (by clicking the **In** check box for `train01` in the **Ligands** table).

To do this, you must close the **Edit Features** dialog box, then reopen it when the change is made. The Workspace displays the ligand `train01` and its acceptor features.

5. In the **Edit Features** dialog box, click the **Mark** box next to the SMARTS pattern `[O;X1]=[C,c]`, which is about half-way down the **Pattern** list table.

The C=O bond in the carboxylic acid in `train01` is now marked. However, there is no pink sphere on the oxygen atom, so it is evidently not being perceived as an acceptor feature. This is because the built-in definitions assume that the user will *not* want to treat this as an acceptor due to the fact that the `-COOH` group is likely to be ionized. Accordingly, there is an *excluded* pattern further down in the table that matches this type of oxygen and excludes it from being flagged as an acceptor.

6. Clear the **Mark** box next to the SMARTS pattern `[O;X1]=[C,c]`.

The C=O bond in `train01` is no longer marked.

7. Scroll down the table until you find the SMARTS pattern `O=C[O-,OH]`.

The **Exclude** box is checked, which means this is an excluded (NOT) pattern.

8. Click the **Mark** box next to this pattern.

The O=C–O moiety is now marked on `train01`.

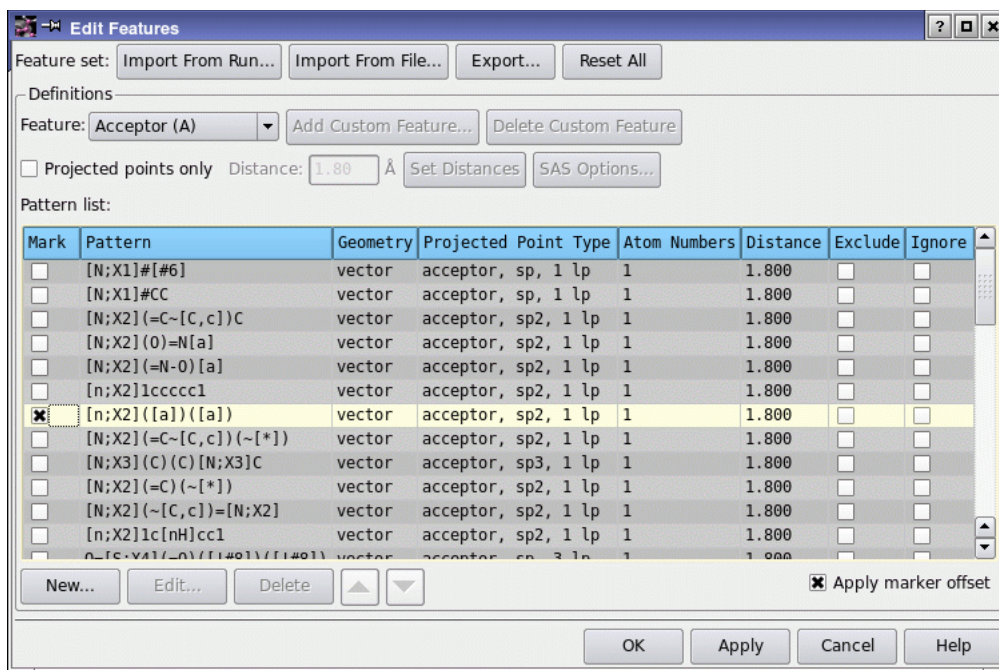


Figure 2.5. The Edit Features dialog box.

Since the Geometry of this pattern is point (that is, the feature is located on a single atom), it excludes only features that are also located on a single atom, that is, features with point and vector geometries. There is no entry in the Atom Numbers column, so the location of the feature defaults to the first atom in the pattern, which is the double bonded oxygen. So this exclude pattern matches all $\text{O}=\text{C}-\text{O}$ moieties, and prevents the double bonded oxygen in $-\text{COOH}$ from being flagged as an acceptor by any other pattern. As noted previously, excluded patterns are actually processed first, so the double bonded oxygen would be flagged for exclusion before any of the regular acceptor patterns were applied.

For the AT_1 data set, almost all of the ligands have an acceptor feature coming off the imidazole ring at the same position as the $-\text{COOH}$ group in *train01*, indicating that there is probably an important hydrogen-bonding interaction going on. Moreover, *train01* is the most active ligand in the data set and it differs from the second most active ligand (*train02*) only by replacement of a $-\text{CH}_2\text{OH}$ group with $-\text{COOH}$. It would appear, then, that $-\text{COOH}$ strengthens this hydrogen bonding interaction. Therefore, when looking for common pharmacophore hypotheses, it may be wise to allow at least one of the oxygens in this $-\text{COOH}$ to act as an acceptor. Accordingly, you will *ignore* this exclude definition, so that it is not applied.

9. In the Ignore column, click the box for the O=C[O- , OH] pattern.

The ignore operation is equivalent to removing the definition from the table, although you can always reinstate it by clearing the Ignore check box.

10. Click Apply at the bottom of the panel to update the definitions.

A delay occurs as the new feature definitions are applied to the first conformation of each ligand. When this process is complete, a pink acceptor sphere appears on the oxygen in the C=O moiety. The excluded pattern definition O=C[O- , OH] is now being ignored, so that -COOH groups will be perceived as containing an acceptor feature.

It is important to note that the built-in definitions used to identify hydrophobic features (H) and aromatic rings (R) are not based on SMARTS patterns. Rather, special algorithms are applied to detect these features automatically, and more efficiently, than would be possible using SMARTS patterns. However, it is possible to add new SMARTS-based definitions to any of the feature types, and to add new features with their own definitions. These possibilities are explored in the next two optional exercises. If you do not want to do these exercises, click OK in the Edit Features dialog box and skip to [Section 2.9 on page 18](#).

2.7 Adding New Feature Definitions

The built-in feature definitions are reasonable, but there may be times when you need to add your own definitions to identify features that aren't accounted for by the built-in set. In this exercise you will define a new type of acceptor feature. The feature you will add is an aromatic ring, which can function as a weak acceptor.

Note: This exercise is optional, and does not affect the final pharmacophore model.

1. Select (A) Acceptor in the features list.

The acceptor features are displayed in the Workspace.

2. In the Edit Features dialog box, select Acceptor (A) from the Feature option menu.
3. Click the first row in the Pattern list table.
4. Click New.

The New Pattern dialog box is displayed.

5. In the SMARTS pattern text box, type c1cccc1.
6. Choose Group from the Geometry option menu.

The geometry in this case is defined by a group of atoms that together contribute an acceptor rather than a single atom.

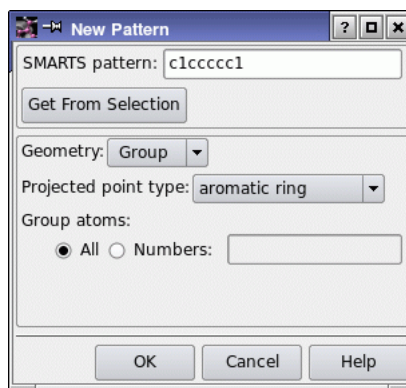


Figure 2.6. The New Pattern dialog box.

7. Choose aromatic ring from the Projected point type option menu.

Most acceptors have a directionality that can be precisely defined by considering the orientation of the acceptor lone pairs with respect to some plane in which the atom lies. In this case the direction is perpendicular to the ring.

8. Click OK.

A new acceptor pattern appears at the top of the Pattern list table. In this position in the table, this pattern is matched before all others. You could move this new pattern vertically in the table using the arrowhead buttons, but doing so ultimately has no effect on the feature perception since there is no other similar functional group.

9. Click the Mark box for this new definition.

The two occurrences of aromatic rings are marked in the Workspace.

10. Click Apply at the bottom of the panel to update the definitions.

After a short delay, a pink sphere appears on each of the rings, indicating that they will be treated as an acceptor. This was for demonstration purposes only, so we should remove the definition before proceeding.

11. Select the newly created pattern in the Pattern list table.
12. Click the Delete button below the table to remove the definition permanently.
13. Click Apply at the bottom of the panel.

The rings are no longer perceived as acceptors. You could have accomplished the same thing by simply *ignoring* the new definition, without permanently deleting it.

2.8 Adding New Feature Types

It is also possible to create an entirely new type of feature (X, Y or Z) beyond the set of built-in feature types (A, D, H, N, P, R). In this exercise, you will create a feature that identifies aromatic rings as weak H-bond acceptors. In practice, you might want to create feature types for weak acceptors and strong acceptors so that the two can be separated in a pharmacophore model. Here, to define the SMARTS pattern, you will use Workspace selection.

Note: This exercise is optional, and does not affect the final pharmacophore model.

1. In the Edit Features panel, choose Custom (X) from the Feature option menu.

The Pattern list table displays three default features that are not defined in terms of SMARTS patterns. The Ignore column is checked for each of these features, so they will not be used unless you clear the checks this column.

2. In the Workspace, select the six carbon atoms in an aromatic ring.

You can do this using shift-click or by dragging over the atoms.

3. Click the New button below the Pattern list table.

The New Pattern dialog box is displayed.

4. Click Get From Selection.

A SMARTS pattern for the selected atoms appears in the SMARTS pattern text box.

5. Ensure that Geometry is Group, Projected point type is aromatic ring and Group atoms is All.

6. Click OK.

The new pattern appears in the Pattern list table.

7. Check the Mark box next to the new pattern.

The aromatic rings are marked in the Workspace.

8. Click Apply at the bottom of the Edit Features dialog box.

9. In the Develop Pharmacophore Model panel, ensure that (X) Custom is selected in the list under Show markers for selected features to display occurrences of this new feature.

After a brief delay, a turquoise sphere with arrows appears at the centroid of each ring, indicating that the new X feature type has been perceived.

This example was for demonstration purposes only, so you should remove the custom feature X before proceeding.

10. In the Edit Features dialog box, ensure that the Custom (X) definitions are displayed.
11. Click Delete Custom Feature.

The Custom (X) definitions are removed and the Acceptor (A) definitions are displayed.

12. Click OK.

The Edit Features panel is closed and the updated definitions are stored.

13. Verify that the custom X feature is no longer visible.

You are now ready to apply the feature definitions to create pharmacophore sites for all conformations.

2.9 Creating Pharmacophore Sites

All of the operations done thus far applied the feature definitions only to the first conformation of each ligand. Before you can proceed to the next step in the workflow, you must apply the feature definitions to all conformations. Pharmacophore sites are created for all entries in the Ligands table, regardless of whether any rows are selected.

1. Click the Create Sites button near the top of the panel (not the one in the Guide).
2. Click Start in the Start dialog box.

This job requires less than a minute on a 2 GHz Pentium 4 processor. Incorporation of results does not add any new information to the Ligands table, but the Find Common Pharmacophores button at the bottom of the panel is now active, and you can proceed to the next step in the workflow.

2.10 Proceeding to Find Common Pharmacophores

In this part of the workflow, pharmacophores from all conformations of the active ligands are examined, and those pharmacophores that contain identical sets of features with very similar spatial arrangements are grouped together. If a given group is found to contain at least one pharmacophore from each active ligand, then this group gives rise to a *common pharmacophore*. Any single pharmacophore in the group could ultimately become a common pharmacophore *hypothesis*.

- Click either Find Common Pharmacophores or Next.

The Find Common Pharmacophores step is displayed.

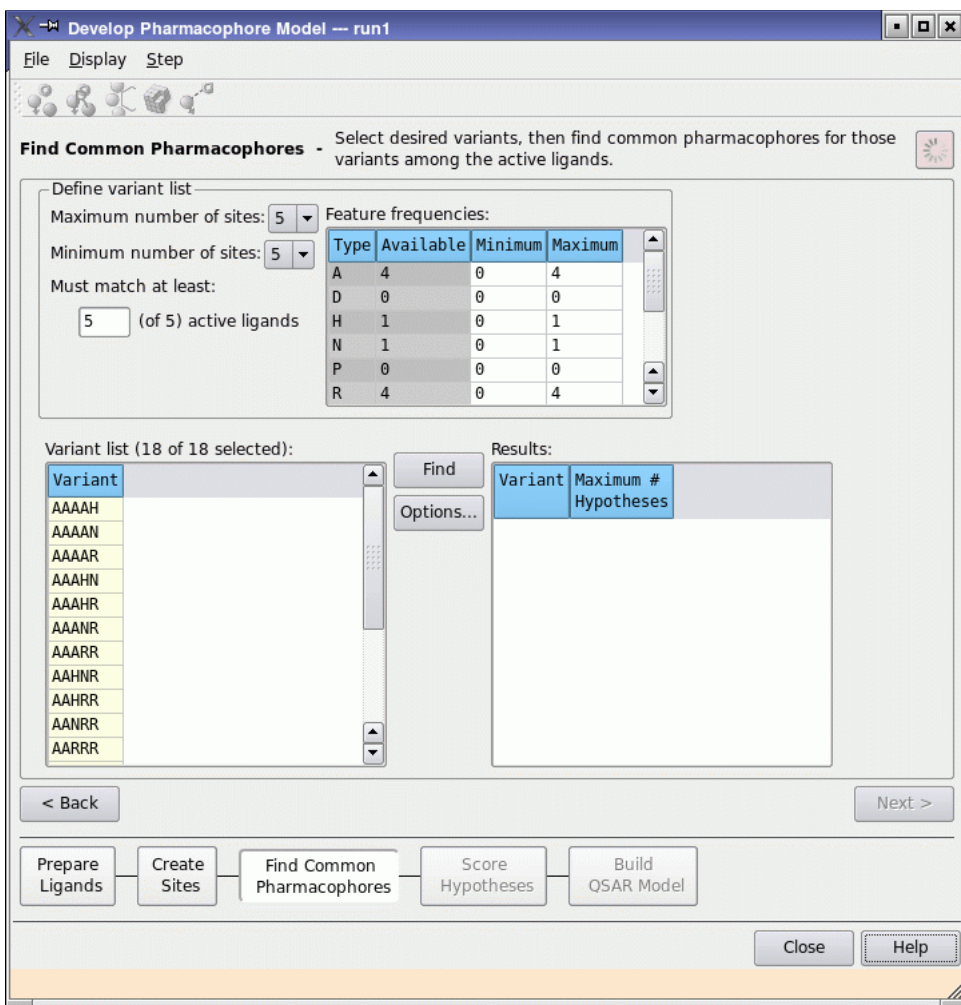


Figure 2.7. The Find Common Pharmacophores step.

By default, Phase looks for common 5-point pharmacophores, that is, pharmacophores containing 5 sites. The maximum and minimum number of sites can be set to any value from 3 to 7 inclusive. If the number of sites is too large, you might not find any common pharmacophores, but if the number of sites is too small, the common pharmacophores might not contain all required features, and therefore might not discriminate between actives and inactives very well. Phase searches for common pharmacophores starting from the maximum number and decreasing the number of sites until it finds common pharmacophores. It is therefore usually a good idea to start with a high maximum number and set the minimum to the minimum acceptable number.

Also by default, Phase looks for pharmacophores that are common to all active ligands. However, Phase allows you to relax this criterion so that a common pharmacophore need only match a subset of the active ligands. This is often a necessity when the actives are highly diverse. So in general, a common pharmacophore must match a *minimum required number of actives*, where that minimum number is set by the user. The present set of five actives is not very diverse, so leave the value in the Must Match at least option at 5.

In the Feature frequencies table, there is a column labeled Available. This is the upper limit on the number of times each feature could possibly be found in any common pharmacophore. For example, the table indicates that no more than one negative feature (N) can appear in any common pharmacophore. This value is arrived at by identifying the active ligand with the smallest number of negative features. If the minimum required number of actives is reduced, then the Available values may increase, depending on whether the most feature-deficient ligands could be excluded from the subset.

The Minimum and Maximum columns in the Feature frequencies table contain user-settable limits on the number of times a feature is *allowed* to appear in a common pharmacophore. For example, if we wanted common pharmacophores to contain at least one, but no more than three acceptors (A), we would change the Minimum and Maximum limits on this feature to 1 and 3.

A list of 19 *variants* also appears on the panel, reflecting the 19 possible combinations of features that could give rise to common pharmacophores. For example, AAARR indicates that the potential exists for common pharmacophores containing three acceptors and two aromatic rings. The feature combinations you see here are determined entirely by the Minimum and Maximum limits in the Feature frequencies table.

2.11 Changing the Number of Sites in Common Pharmacophores

This exercise examines how the features frequencies and variants change with the number of sites in the common pharmacophores we are searching for.

1. Choose 4 from the Maximum number of sites option menu.

The only change in the Feature frequencies table is that the Maximum allowed frequency of acceptors has been reduced from 5 to 4. This merely reflects the fact that the common pharmacophores we are looking for cannot contain more than 4 sites. The number of variants has dropped from 19 to 16, and of course there are only 4 features in each variant.

2. Change the maximum number of sites to 6.

The Feature frequencies table is identical to that observed when the maximum number of sites was 5, and there are once again 19 variants.

3. Change the maximum number of sites to 7.

The Feature frequencies table is unchanged, but the number of variants drops to 16. Combinatorics dictates that there will be a peak in the number of variants as the number of sites is varied, and we see that the peak occurs at 5 and 6 sites. You will be searching for common 6-point pharmacophores.

4. Change the maximum number of sites back to 6.
5. Change the minimum number of sites to 6.

2.12 Changing the Allowed Feature Frequencies

The Feature frequencies table and the variant list indicate that we can search for common pharmacophores with up to five acceptors. However, it is very unlikely that a ligand would bind to the receptor through five, or even four, hydrogen-bond acceptor interactions. Here, some chemical intuition is needed to filter out results that are scientifically unsound or unlikely.

1. In the Feature frequencies table, change the minimum and maximum number of acceptors (A) to 1 and 3, respectively.

The number of variants drops from 19 to 11 and each variant contains between 1 and 3 acceptor features.

We know that each active ligand contains a tetrazole ring and therefore at least one negative feature. The tetrazole has no doubt been put there for a good reason, so it is pretty safe to assume that the hypothesis should contain a negative feature.

2. In the Feature frequencies table, change the minimum number of negative features (N) from 0 to 1.

The number of variants drops from 11 to 6, and each variant contains a negative feature.

The Feature frequencies table also indicates that there could be as many as 4 aromatic rings. However, one of these rings is the tetrazole, which we presume will be acting as a negative feature, so it is not necessary to consider so many aromatic rings.

3. Change the maximum number of aromatic rings (R) from 4 to 3.

The number of variants drops from 6 to 5, and no variant contains more than 3 aromatic rings.

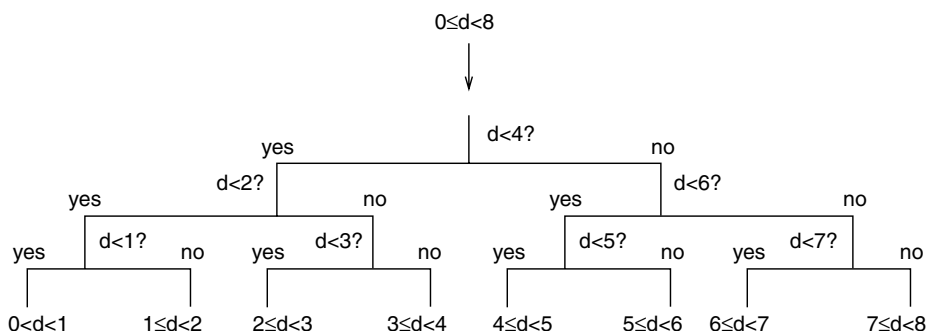


Figure 2.8. Binary decision tree.

2.13 Examining Options for Finding Common Pharmacophores

To understand the options for finding common pharmacophores, we must first examine how the underlying algorithm works. Common pharmacophores are identified using a tree-based partitioning technique that groups together similar pharmacophores according to their *intersite distances*, i.e., the distances between pairs of sites in the pharmacophore. Accordingly, each k -point pharmacophore is represented by a vector of n distances, where $n = k \cdot (k-1)/2$. Each intersite distance d is filtered through a binary decision tree, such as in Figure 2.8.

Note: This exercise is optional, and does not affect the final pharmacophore model.

The tree in Figure 2.8 has a depth of three and partitions distances (in angstroms) on the interval $0 < d \leq 8$ into bins that are 1 Å wide. If each of the n distances in a pharmacophore is filtered in this manner, an n -dimensional partitioning of the pharmacophore is created.

This representation is referred to as an n -dimensional box, where the sides of the box are equal to the bin width. Thus a pharmacophore is mapped, according to its intersite distances, into a box of finite size. All pharmacophores that are mapped into the same box are considered to be similar enough to facilitate identification of a common pharmacophore. So if each of a minimum required number of actives contributes at least one pharmacophore to a particular box, then that box represents a common pharmacophore. Such boxes are said to *survive* the partitioning procedure, while all others are eliminated.

1. Click Options.

The Find Common Pharmacophores - Options dialog box is displayed. Most of these options can be directly linked to the example tree in Figure 2.8.

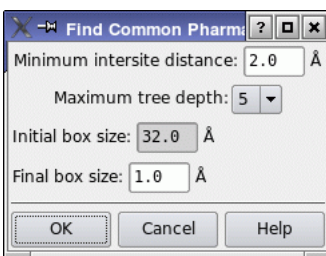


Figure 2.9. The Find Common Pharmacophores - Options dialog box.

Initial box size refers to the width of the initial distance interval. By default this is 32 Å, indicating that intersite distances on the interval $0 < d \leq 16$ will be processed. Final box size refers to the size of the n -dimensional boxes into which the intersite distances are ultimately mapped. The default is 1 Å. Note that Initial box size is not actually set by the user, but rather computed from Maximum tree depth and Final box size:

$$\text{Initial box size} = (\text{Final box size}) \times 2^{(\text{Maximum tree depth})}$$

2. To verify the relationship among Initial box size, Final box size and Maximum tree depth, change Maximum tree depth to 4.

Initial box size decreases to 16.0.

The only remaining option is Minimum intersite distance. This parameter determines whether or not a pharmacophore will be rejected because a pair of features is too close together. For example, suppose a ligand contains a $-\text{CH}_2\text{OH}$ group. Using the built-in feature definitions, an acceptor feature would be placed on the oxygen and a donor feature would be placed on the hydrogen. While it is theoretically possible for an oxygen atom to act as both donor and acceptor (e.g., water), it is probably unlikely in a ligand-receptor interaction. Accordingly, we would typically want to ignore pharmacophores that contain both the acceptor and donor features associated with a single oxygen atom. Since these two features are separated by only about 1 Å, a Minimum intersite distance of 2.0 Å would cause the entire pharmacophore to be rejected before it is even run through the partitioning tree. The default value of 2.0 Å is recommended, as are all other defaults in this dialog box.

3. Click Cancel to close this dialog box and keep all default settings.

You are now ready to search for common pharmacophores.

2.14 Finding Common Pharmacophores

A total of five variants are examined to identify 6-point pharmacophores that are common to all five of the active ligands.

1. Ensure that all five variants are selected, then click Find.
2. Click Start in the dialog box.

This job requires about two minutes on a 2 GHz Pentium 4 processor. After the job incorporates, the Results table shows that only two of the five variants yielded common pharmacophores. Maximum # Hypotheses is the number of different 1-Å boxes that survived the partitioning procedure, and hence the number of distinct common pharmacophores that were identified for each variant. Recall that a given box contains one or more pharmacophores from each of the minimum required number of active ligands. Exactly one pharmacophore from each box will be selected as a potential hypothesis, and this selection takes place in the Score Hypotheses step.

2.15 Proceeding to Score Hypotheses

In the next step, common pharmacophores are examined, and a scoring procedure is applied to identify the pharmacophore from each box that yields the best alignment of the active ligands. This pharmacophore provides a hypothesis to explain how the active ligands bind to the receptor. There will of course be many hypotheses, because there are many boxes. The scoring procedure provides a ranking of the different hypotheses, allowing you to make rational choices about which hypotheses are most appropriate for further investigation.

- Click either Score Hypotheses or Next.

The Score Hypotheses step is displayed. The Hypotheses table is empty because scoring has not yet been done.

2.16 Scoring Hypotheses

In this exercise, you will examine settings that control how hypotheses are selected from each surviving box and how they are ranked with respect to one another, and then run scoring jobs to examine the effect of including various terms in the score.

1. Click Score Actives.

The Score Actives dialog box is displayed.

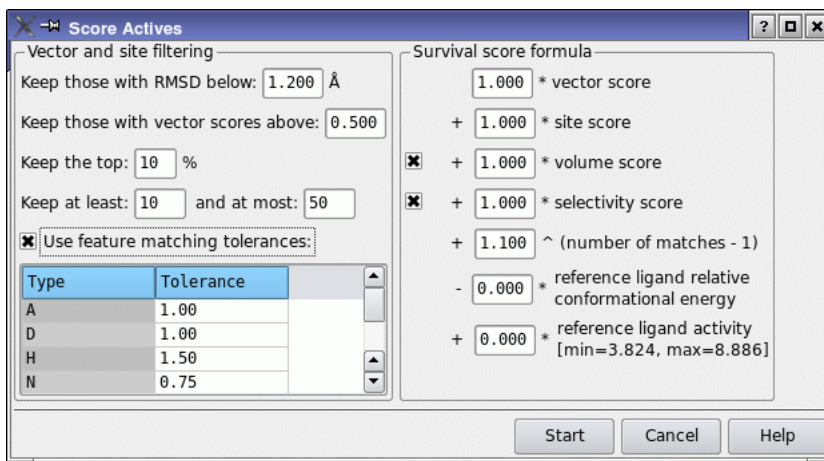


Figure 2.10. The Score Actives dialog box.

To understand the options in this dialog box, you must first understand the scoring process. A surviving box contains a set of very similar pharmacophores culled from conformations of a minimum number of active ligands, and certain of these ligands may contribute more than one pharmacophore to a box. Each pharmacophore and its associated ligand are treated temporarily as a *reference* in order to assign a score. This means the other *non-reference* pharmacophores in the box are aligned, one-by-one, to the reference pharmacophore, using a standard least-squares procedure applied to the corresponding pairs of site points.

At this stage, the quality of each alignment is measured using up to three terms: (1) the root-mean-squared deviation (RMSD) in the site point positions; (2) the average cosine of the angles formed by corresponding pairs of vector features (acceptors, donors and aromatic rings); and (3) a volume overlay term based on van der Waals models of the non-hydrogen atoms in each pair of structures.

$$S_{\text{vol}}(i) = V_{\text{common}}(i)/V_{\text{total}}(i)$$

$V_{\text{common}}(i)$ is the common or overlapping volume between ligand i and the reference ligand, while $V_{\text{total}}(i)$ is the total volume occupied by both ligands. The volume term is computed only if the option to score by volume is selected. These two or three terms are combined with separate weights to yield a combined alignment score for each non-reference pharmacophore that has been aligned to the reference. If a non-reference ligand contributes more than one pharmacophore to the box, the pharmacophore yielding the best alignment to the reference is selected. The overall multi-ligand alignment score for a given reference pharmacophore is the average score from the best set of individual alignments.

In principle, a reference pharmacophore could yield a good average score, even though it contains one or two very poor individual alignments. For this reason, user-adjustable cutoffs are applied to the RMSD values and vector cosines of each individual alignment. Any reference pharmacophore that violates a cutoff in any individual alignment is eliminated.

After all pharmacophores in a box have been treated as a reference, the one yielding the highest multi-ligand alignment score is selected as the hypothesis for that box. The ligand that contributes the reference pharmacophore is referred to as the *reference ligand* for that hypothesis. Note that the non-reference information is carried along with each hypothesis so that additional scoring may be performed using the optimal multi-ligand alignment.

Once hypotheses have been identified across all boxes, you may want to eliminate some of the lower scoring ones. A percentage cutoff may be applied to the overall alignment score so that, for example, only hypotheses in the top 10% are retained. You may also request that some minimum number of hypotheses be retained, just in case the percentage filter yields a very small number.

Once this stage of scoring is completed, you can further refine the ranking of the hypotheses by adding to the scoring function terms for selectivity, reference ligand relative energy, and reference ligand activity. You might also want to weight hypotheses that match more actives.

Selectivity is an empirical estimate of the *rarity* of a hypothesis, i.e., what fraction of molecules are likely to match the hypothesis, regardless of their activity toward the receptor. Selectivity is defined on a logarithmic scale, so a value of 2 means that $1/10^2$ molecules would be expected to match the hypothesis. Higher selectivity is desirable because it indicates that the hypothesis is more likely to be unique to the active ligands. Selectivity is only a rough estimate of the rarity, so you should be careful not to place too much emphasis on it in the overall ranking of hypotheses. As with the other types of scores, selectivity can be added to the overall score with its own adjustable weight.

You may also wish to assign higher scores to hypotheses that match a greater number of active ligands. This is relevant when the required minimum number of actives is smaller than the total number of actives. The reward comes in the form of w^m , where w is adjustable (1.0 by default) and m is the number of actives that match the hypothesis. If w is increased above 1.0, care must be taken not to make it too large, or it may completely dominate the scoring function.

Logically, the best hypothesis should have as a reference ligand the ligand that has the highest activity. You can weight the score towards this goal by including the activity in the scoring.

If the conformation of the reference ligand is high in energy with respect to the lowest conformation, this can indicate a poor hypothesis, because the internal strain of the ligand must be overcome when the ligand binds. You can include a penalty term by subtracting off a multiple of the reference ligand relative energy.

By default, none of these optional scores contribute to the ranking of the hypotheses. Although selectivity score is selected as part of the scoring function, its weight is zero. However, all individual components of the score are reported, so you can always consider selectivity, if you so desire, when making choices among hypotheses that have very similar overall scores.

2. Check that the weights of the selectivity score and the number of matches are 0.0 and 1.0.
3. Click **Start**, and click **Start** again in the **Start** dialog box.

This job requires less than one minute on a 2 GHz Pentium 4 processor. Once the job is incorporated, the **Hypotheses** table contains information about the highest scoring hypotheses from each variant.

Note that the range of scores among the reported hypotheses is only 0.5, so none of the hypotheses is particularly poor.

Next, you will check whether any of these hypotheses can be eliminated based on its match to the inactives.

4. Click **Score Inactives**.

The **Score Inactives** dialog box opens. Leave the inactives weight at its default value of 1.0.

5. Click **Start**, and click **Start** again in the **Start** dialog box.

The job takes less than a minute. When it finishes, the **Survival-inactive** column of the **Hypotheses** table is populated with values.

The survival score is reduced fairly evenly for all the hypotheses. A reduction of one unit is expected, because the scoring function for the inactives is the same as for the actives, and it includes a contribution of 1.0 for the number of matches. The remainder of the reduction is fairly uniform and fairly small, indicating that the inactives do not match the hypotheses well enough to eliminate any of them. On this basis, we can be fairly confident that the hypotheses are reasonable explanations of the activity.

Finally, you will adjust the scoring to take account of the activity. Of the five actives used for model development, one has a significantly higher activity than the others. Adding an activity reward could change the ranking of the hypotheses that have this ligand as the reference.

6. Click **Rescore**.

The **Rescore Hypotheses** dialog box opens.

7. Enter 0.3 in the reference ligand activity text box.

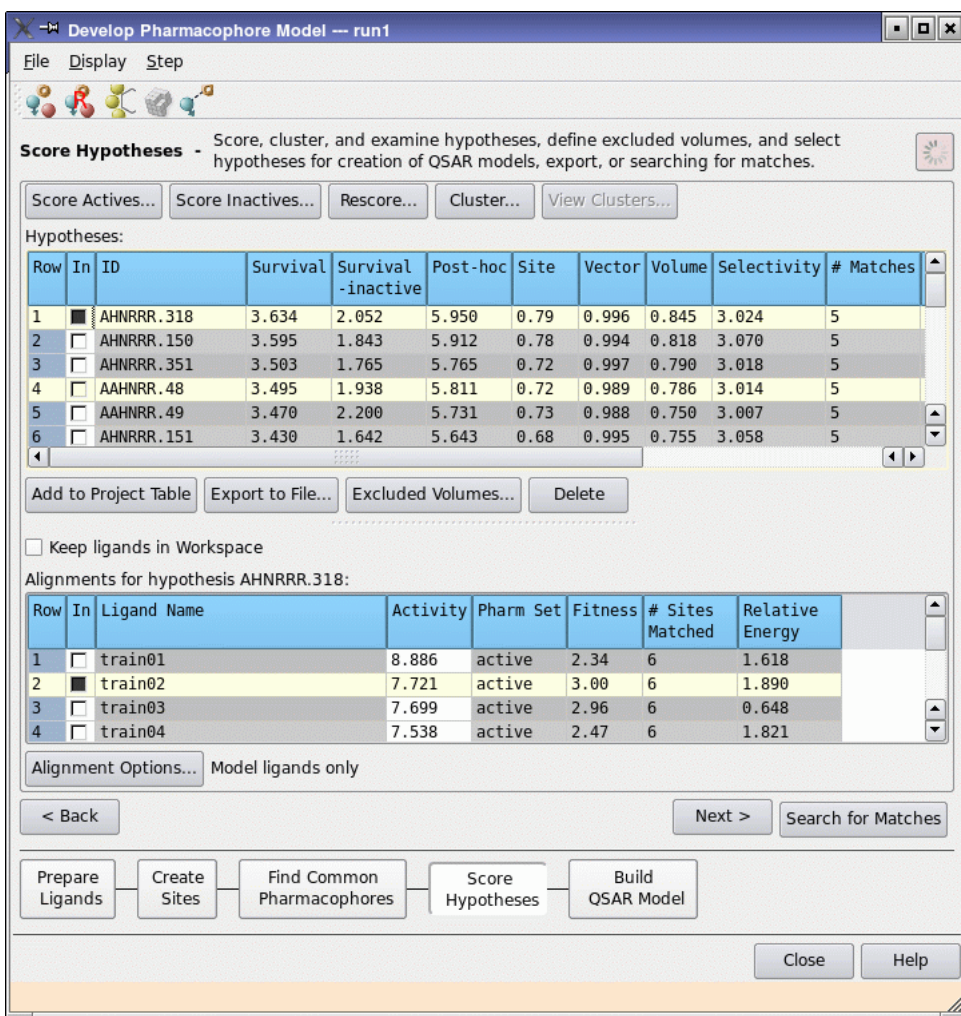


Figure 2.11. The Score Hypotheses step after scoring.

- Click OK.

The results are returned almost immediately, in the Post-hoc column. Scoring the actives and the inactives requires alignment of the ligands, which takes a little time. For rescoreing the alignment is already done.

- Click twice on the Post-hoc column heading, to sort the hypotheses in descending order by this score.

Now, two other hypotheses are at the top, followed by the set of hypotheses with the top survival scores. In the next exercise, you will examine some of these hypotheses and the ligand alignments.

2.17 Viewing Hypotheses and Ligand Alignments

In this exercise you will examine the nature of the top-scoring hypotheses and the quality of the associated ligand alignments.

1. Sort the Hypotheses table by survival score, by clicking twice on the Survival column heading.

The top scoring hypothesis comes from an AHNRRR variant, AHNRRR.318 (The numeric suffix is merely the index of the box from which the hypothesis came.) Among the top few hypotheses is an AHNRRR variant that has scores that are close to those of the top scoring hypothesis. As we shall see, this is due to the fact that both hypotheses align the active ligands in very similar ways.

2. Click the In column in the first row in the Hypotheses table.

The features of the hypothesis are displayed in the Workspace. If the hypothesis is *not* visible in the Workspace, click on the Display Hypothesis toolbar button at the top of the Develop Pharmacophore Model panel.



The Alignments table is filled with a record for each ligand. The records for the ligands not used for the model (the *non-model* ligands) are dark gray, indicating that no alignment was done for these ligands. The records provide information on the conformation whose pharmacophores yielded the best multi-ligand alignment to the hypothesis when it was selected as the reference from its box. Fitness measures the quality of each alignment using a weighted sum of alignment and volume scores, just as in the total scoring function. Note that the second row of the Alignments table is blue, indicating that `train02` is the reference ligand, i.e., the ligand from which the hypothesis came. Its fitness score is a perfect 3.0—the maximum possible score with the scoring options that were selected—because it corresponds to the alignment of `train02` onto itself.

3. Place the reference ligand in the workspace by clicking the appropriate In box in the Alignments table.

`train02` is overlaid onto the hypothesis. The features of the hypothesis are perfectly positioned on the matching sites of this ligand, because the hypothesis comes from the reference ligand, so its features must coincide with those of the reference ligand.

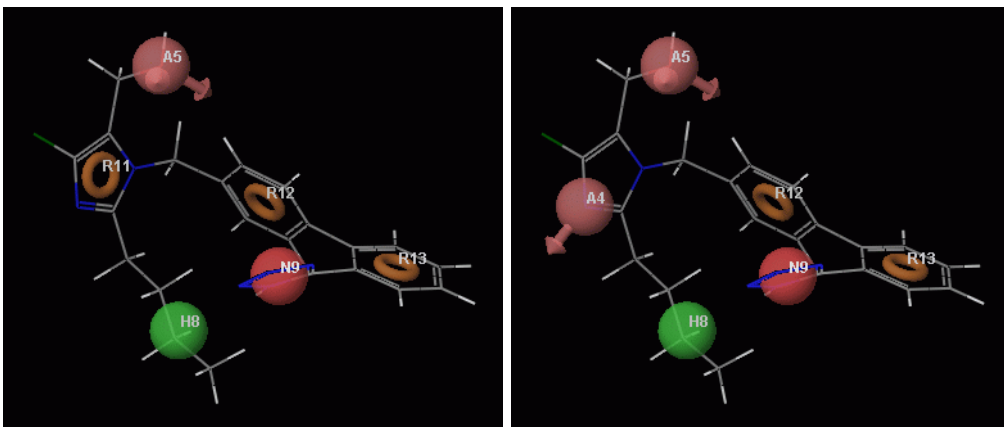


Figure 2.12. Hypotheses AHNRRR.318 (L) and AAHNRR.48 (R) with reference ligand train02.

4. Include train03 in the Workspace by control-clicking its In box.

train03 is overlaid onto the hypothesis by aligning its 6-point pharmacophore with the six features in the hypothesis. Observe that the train03 and train02 ligands are only slightly out of alignment with each other.

5. Include the remaining actives in the Workspace.

Most ligands superimpose on each other with only minimal offset. This should come as no surprise because most of the Fitness scores are close to 3.0. In fact, ligands train02, train03, and train04 only differ by the substitution of Cl or I for H. The exception is train01, whose fitness score is about 2.3. If you place this ligand in the Workspace with the reference ligand, you can see the deviation in the alignment.

6. Examine the hypothesis AHNRRR.150 by selecting its row in the Hypotheses table.

The Alignments table is updated to reflect the best alignments for AHNRRR.150. As was for AHNRRR.318, train02 is the reference ligand, but the conformation of the reference ligand is different, as indicated by the values in the Energy column of the Hypotheses table.

7. Examine the hypothesis AAHNRR.48 by selecting its row in the Hypotheses table.

Once again, train02 is the reference ligand. The side-by-side view of train02 aligned to AHNRRR.318 and AAHNRR.48 is shown in [Figure 2.12](#).

It should be evident that the same reference ligand conformation was selected in both cases, and that the two hypotheses differ only in whether the imidazole acts as an aromatic feature or as an acceptor. In situations like this, it is difficult to assess which

hypothesis is most reasonable, so one often pursues both for further investigation. For this tutorial, we shall examine both AHNRRR.318 and AAHNRR.48.

In the next step you will build a 3D QSAR model, which requires alignments for ligands outside the active set. The 49 training and test set ligands cannot all be expected to match all six sites in the hypothesis. Some of the weak binders will be missing one or more features contained in the hypothesis. To deal with this possibility, Phase uses *partial matching* to obtain alignments for these ligands. If at least three sites in the hypothesis can be matched, an unambiguous alignment is obtained. For each ligand outside the active set, then, Phase searches for matches involving the largest possible number of sites, and identifies the match that yields the highest fitness score. Alignments can give information about which features are important and which are not, especially for actives that are not in the training set.

8. Select AAHNRR.48 in the hypothesis table.
9. Click Alignment Options below the Alignments table.

The Alignment Options dialog box opens. In addition to selecting the option to align the “non-model” ligands (those that were not in the pharm set or did not match all sites in the hypothesis), you can require certain features to match when the alignment is performed. If the features do not match for a particular ligand, the ligand is not aligned. For this tutorial we will not require any matches.

10. Select Align non-model ligands, and click OK.

After a short delay, the Alignments table contains an entry for all ligands. It is not always possible to obtain an alignment for every ligand, but it has happened in this case. Scrolling down the table, you will find matches involving 4, 5, and 6 sites and a wide range of fitness values.

Now you will examine the alignments for several ligands: train12, which matches all 6 points but has a poor fitness score, train25, which matches 4 points and is inactive, and test02, which only matches 5 points but is highly active.

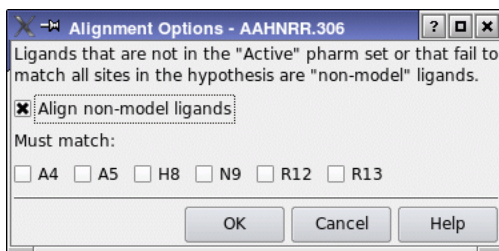


Figure 2.13. The Alignment Options dialog box.

11. Place `train12` in the Workspace.

The features in `train12` do not align well with the corresponding features of the hypothesis. This does not necessarily mean that `train12` could never achieve better alignment, but rather that the conformations we are working with contain no such alignment. However, an examination of the structure shows that a better alignment is unlikely, and there is some internal strain in achieving the current alignment: the energy of this conformer relative to the lowest is more than 4 kcal/mol. When building a QSAR model, it is a good idea to take into account the quality of the superimpositions, especially for members of the training set. Cleaner superimpositions usually yield models with greater statistical significance and greater predictive ability.

12. Place `train25` in the Workspace.

It is immediately obvious that, although the four features that do match align well, there are two features missing. It is reasonable to suppose that at least one of these features is critical to activity.

13. Place `test02` in the Workspace.

This is a highly active compound in the test set. It only matches on five features, implying that the sixth feature is not really necessary for activity.

2.18 Proceeding to Build QSAR Model

Phase QSAR models are based on partial least-squares (PLS) regression, applied to a large set of binary-valued variables that encode whether or not ligand atoms or ligand features occupy various cube-shaped elements of space. Using the hypotheses `AHNRRR.318` and `AAHNRR.48`, you will develop QSAR models to explain the activity data (`pIC50-Exp`), then apply these models to make activity predictions for the test set ligands. Although these hypotheses are the top-scoring hypotheses, there is no necessary connection between the score and the quality of the QSAR model. When you build QSAR models, you should try several hypotheses to ensure that you have a good model. Phase permits you to build models for multiple hypotheses simultaneously.

1. Select `AHNRRR.318` and `AAHNRR.48` in the Hypotheses table.
2. Click either **Build QSAR Model** or **Next**.

The **Build QSAR Model** step is displayed.

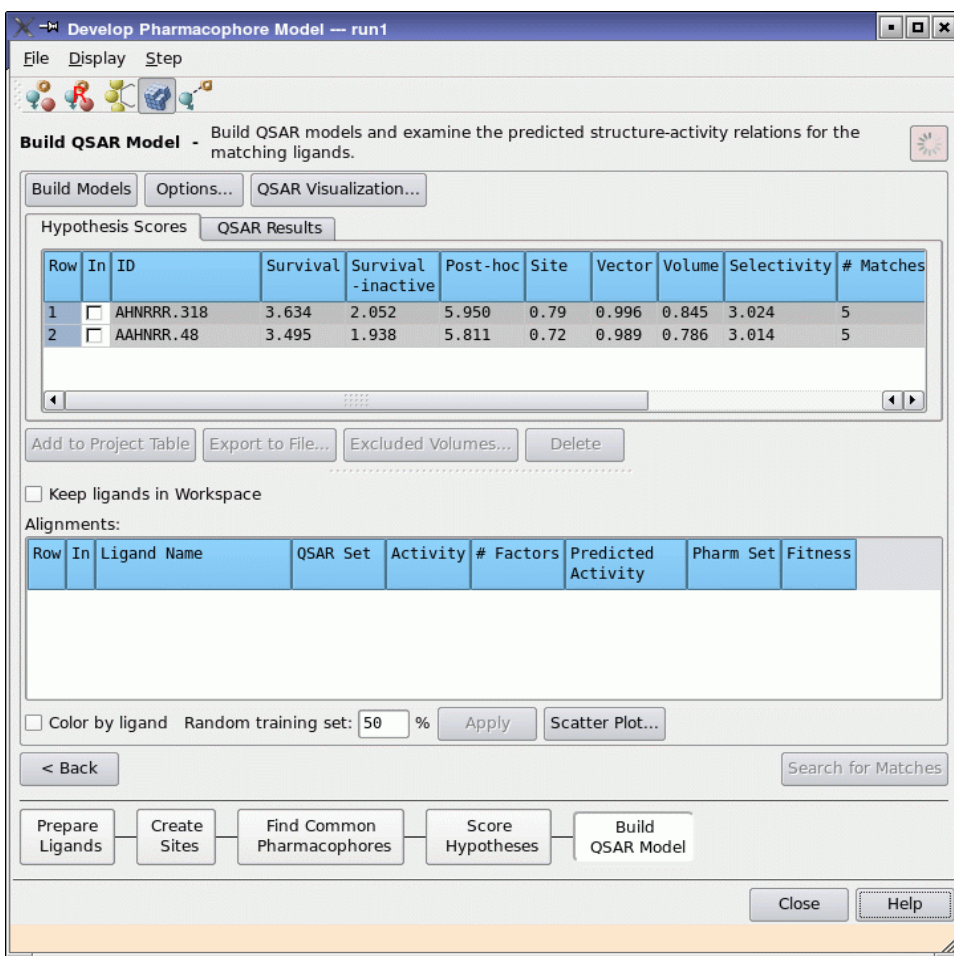


Figure 2.14. The initial view of the Build QSAR Model step.

2.19 Assigning Training and Test Set Memberships

By default, all ligands are placed in the training set, so you must separate them into the proper training and test sets. You need only do this for one hypothesis, because the set membership is the same for all hypotheses in this step.

1. In the Hypothesis Scores table, click the In column for AHNRRR.318.

The hypothesis is displayed in the Workspace, and the Alignments table is populated with data for this hypothesis

2. Select all the test set ligands (test01, test02, etc.) in the Alignments table.

You can do this by clicking test01 and shift-clicking test25. Make sure you do not click in the QSAR Set column. If the ligands are not sorted by name, click the Ligand Name column heading to sort them.

3. Control-click the QSAR Set column for one of the selected ligands, to change the value from training to test.

This results in a training set of 25 ligands and a test set of 24 ligands. If you wanted to eliminate a particular training set (or test set) ligand from consideration, possibly because of a poor superimposition, you could click the corresponding QSAR Set cell until its contents were blank.

Note that the hypothesis is used only to obtain ligand alignments: it does not contribute in any way to the QSAR model itself. But the hypothesis has an *association* with the model, because it defines how ligands should be pre-aligned before applying the model.

2.20 Setting QSAR Model Options

Phase can generate QSAR models that are atom-based or pharmacophore-based. The independent variables in the QSAR model are derived from a regular grid of cubic volume elements that span the space occupied by the training set ligands. Each ligand is represented by a set of bit values (0 or 1) that indicate which volume elements are occupied by either a van der Waals model of the atoms in that ligand or by pharmacophore features of that ligand. In the atom-based model, to distinguish different atom types that occupy the same regions of space, a given cube in the grid may be allocated as many as six bits, accounting for six different classes of atoms. Likewise, the pharmacophore features in the hypotheses are represented by bits for each feature type.

In this exercise, you will be developing an atom-based model, and will set parameters to control the sizes of the cubes and the maximum number of PLS factors to include in the model.

1. In the Develop Pharmacophore Model panel, click Options.

The Build QSAR Model options dialog box is displayed.

2. Set Grid spacing to 1.0.
3. Set Maximum PLS Factors to 3.
4. Under Model type, ensure that Atom-based is selected.

Note that for the pharmacophore-based models, you can adjust the feature radii.

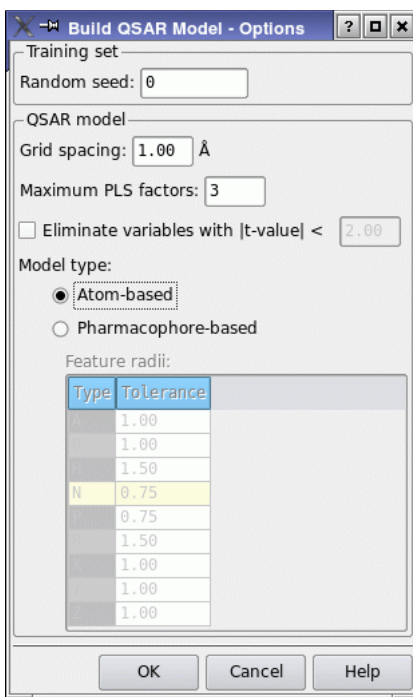


Figure 2.15. The Build QSAR Model - Options dialog box.

5. Click OK.

The cubes that define the independent variables will be 1 Å on each side, and atom-based linear regression models will be built containing one, two and three PLS factors.

2.21 Building the QSAR Models

One-factor, two-factor and three-factor PLS regression models are created using the training set ligands and then applied to the test set ligands.

1. Click Build Models.

The Start dialog box opens.

2. Select a host, and click Start.

The job status icon at the top right turns green and spins. When it stops, data for three PLS regression models fills the QSAR results table. Each regression is a QSAR model.

For a regression with m PLS factors, built using n training set ligands, the statistical parameters shown in the tables are:

SD	The standard deviation of regression. This is the RMS error in the fitted activity values, distributed over $n-m-1$ degrees of freedom.
R-squared	The coefficient of determination. A value of 0.80, for example, means that the model accounts for 80% of the variance in the observed activity data. R-squared is always between 0 and 1.
F	The ratio of the model variance to the observed activity variance. The model variance is distributed over m degrees of freedom and the activity variance is distributed over $n-m-1$ degrees of freedom.
P	The significance level of F when treated as a ratio of Chi-squared distributions. A P value of 0.05 means F is significant at the 95% level.
Stability	Stability of the model predictions to changes in the training set composition.
RMSE	The RMS error in the test set predictions.
Q-squared	Directly analogous to R-squared, but based on the test set predictions. Note that Q-squared can take on negative values if the variance in the errors is larger than the variance in the observed activity values.
Pearson-R	Pearson R value for the correlation between the predicted and observed activity for the test set.

It should be apparent that the R-squared value will always increase as the number of PLS factors increases, but the same is not necessarily true of Q-squared.

3. In the QSAR Results table, click the **In** column for AAHNRR. 48.

The hypothesis is displayed in the Workspace, and the Alignments table is populated with data for this hypothesis, including activity predictions for each ligand using one, two and three PLS factors.

Scrolling down to the test set, we see that some predictions are good and some are bad. As might be expected, the prediction for `test02`, the active ligand that matched only five sites in the hypothesis, is significantly lower than experiment.

Of particular interest is `test17`, whose experimental activity value may be in question since its structure was identical to that of `test24`, and which was eliminated when the conformer sets were generated for this tutorial. The 2-factor prediction for this ligand is 4.37, while the experimental activities originally reported for `test17` and `test24` were 5.770 and 4.553, respectively. The first experimental value, 5.770, is the value that was used, so the error in the prediction is quite large. To see how much this observation is skewing the statistics for the test set, you can remove it and rebuild the model.

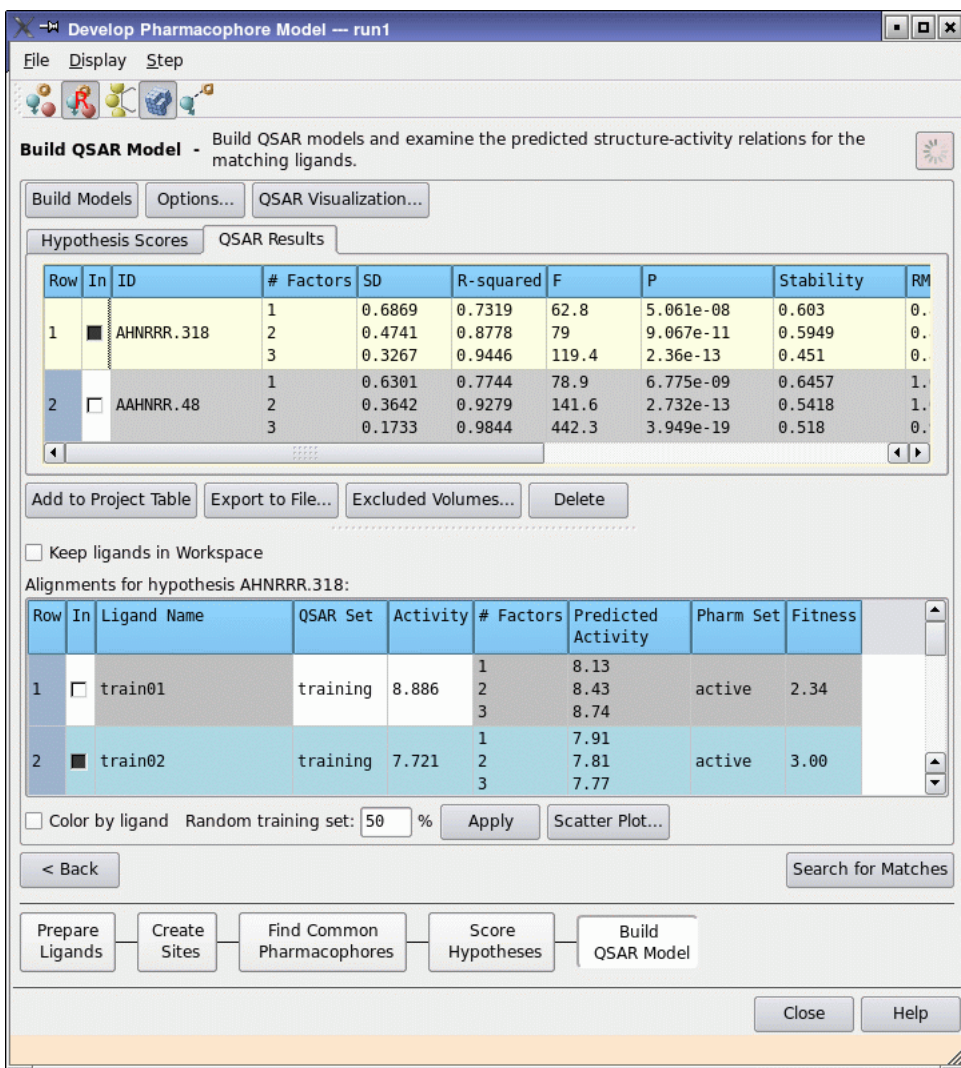


Figure 2.16. The Build QSAR Model step after building models.

- In the Alignments table, click the QSAR Set cell for test17 to change its contents from test to an empty cell.

A dialog box appears warning you that you will invalidate data in forward steps.

- Choose the Proceed option and click OK.

The ligand test17 is no longer a member of the test set.

6. Click Build Models.

The new models are identical to the old models because the training set is the same. However, the Q-squared value for the test set has increased because the poor prediction for test17 is not being considered.

2.22 Visualizing the QSAR Model

Three-dimensional aspects of the QSAR model are examined to help gain an understanding of how the structures of the ligands contribute to the computed activity.

1. Ensure that the View QSAR model toolbar button is selected.



2. Place train01, the most active ligand, in the Workspace.
3. Click QSAR Visualization.

The QSAR Visualization Settings panel is displayed. This panel has various options for displaying characteristics of the QSAR model.

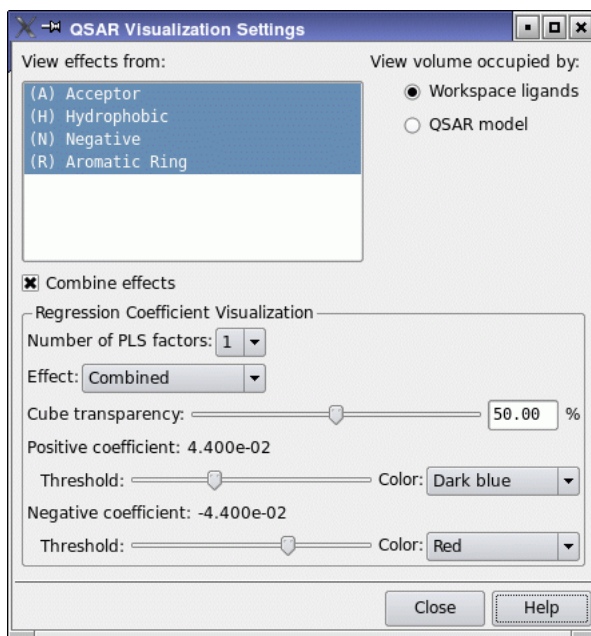


Figure 2.17. The QSAR Visualization Settings panel.

You can view the cubic volume elements occupied by the ligands in the Workspace, or view all the cubes in the QSAR model (i.e., the union of cubes occupied by the 25 training set ligands). You can also view the cubes associated with all atom classes or a specific atom class. The number of PLS factors determines which of the three regression models you are viewing, and the positive and negative coefficient thresholds allow you to see only the cubes whose PLS regression coefficients exceed a particular tolerance.

4. Ensure that all items in the View effects from list are selected, and that Combine effects is selected
5. Ensure that Workspace ligands is selected under View volume occupied by.
6. Select 2 PLS factors.
7. Move the positive and negative coefficient threshold sliders to an intermediate value, such as 0.015 and -0.015.

In the Workspace, you will see many blue cubes and a small number of red cubes. The blue cubes indicate regions that are favorable for activity and the red cubes indicate regions that are unfavorable. Note that you are viewing only those cubes whose regression coefficients exceed the intermediate thresholds we set.

8. Change the positive and negative coefficient thresholds to more extreme values, such as 0.035 and -0.035.

The number of cubes in the Workspace drops significantly because now you are viewing only very significant terms in the model.

9. Change the thresholds back to 0.015 and -0.015.
10. Remove `train01` from the Workspace and replaced it with `train25`, the least active ligand.

The Workspace now contains many more red cubes than blue cubes, indicating a preponderance of unfavorable interactions.

11. Add `train01` back into the Workspace, while keeping `train25` there as well (control-click the `ln` column for `train01`).

You are now viewing the union of the cubes occupied by the two ligands. It should be evident that `train25` fails to occupy much of the favorable blue volume of `train01`. It should also be evident that this volume is associated primarily with the aromatic and negative features that are not matched by `train25`. So while the hypothesis does not contribute in an explicit sense to the QSAR model, it is reflected implicitly in the regression coefficients and volume occupation patterns. Note that a Phase QSAR model may involve regions of space that extend beyond the physical bounds of the hypothesis because the QSAR considers the volume occupied by all atoms in the ligands.

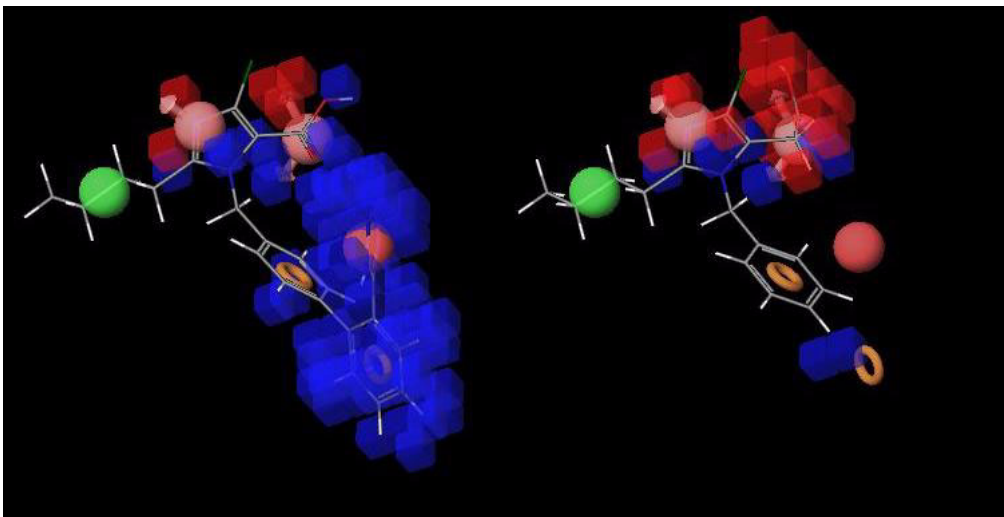


Figure 2.18. QSAR model for ligands train01 and train25.

12. Remove train25 from the Workspace (click train01).

The current view of the model illustrates effects from all atom classes simultaneously, but it is possible to separate out the contributions of individual atom classes.

13. Under View effects from, deselect Combine effects, and select (D) H-bond donor.

This category includes polar hydrogens bonded to nitrogen, oxygen and sulfur.

14. Choose (D) H-bond donor from the Effect option menu.

15. Change the positive and negative coefficient thresholds to lower values, such as 0.003 and -0.003.

The view in the workspace indicates that the tetrazole and carboxylic acid groups of train01 occupy D-type volume, and that this occupation contributes favorably to activity. While we know that both of these groups are likely to be ionized, the QSAR model takes each ligand structure *as is* and assigns each atom to a particular class. So these groups would give rise to negative ionic atoms (type “N”) only if the ligand structures were explicitly ionized. As it is, there are no type N features in the list.

16. Choose (H) Hydrophobic/non-polar in the View effects from list and the Effect option menu.

17. Change the positive and negative coefficient thresholds to 0.015 and -0.015.

Green cubes are now visible throughout much of the structure of `train01`. These are favorable regions occupied by carbons, halogens, and nonpolar hydrogens. Note that the QSAR model does not distinguish between aromatic and nonaromatic carbons.

18. Add `train25` back into the Workspace.

Many purple cubes appear, indicating that `train25` occupies a fair amount of unfavorable volume of type H.

19. Choose (W) Electron-withdrawing in the View effects from list and the Effect option menu.

20. Change the positive and negative coefficient thresholds to lower values, such as 0.003 and -0.003.

The Workspace indicates favorable and unfavorable regions occupied by electron-withdrawing nitrogen and oxygen atoms. W-type volume includes hydrogen bond acceptor atoms, but it does not distinguish cases where a lone pair may not be available due to conjugation, e.g., amide nitrogens.

Experiment with different atom classes and different thresholds to view the various ways in which the QSAR model distinguishes ligands with high and low activities.

Creating a 3D Database

Once a pharmacophore model has been developed, it may be used to search a database, with the goal of identifying additional active molecules. This chapter describes the process of creating Phase 3D databases. You can search a set of 3D structures in a structure file, but if you plan to search the same database with the same feature set multiple times, it is quicker to create a Phase database that includes conformer sets and sites.

If you plan to search the database on a host that is different from the one you will use in this chapter, you must create the database on a file system that is accessible to the other host.

3.1 Creating a New 3D Database

In this exercise you will create a small database from a Maestro file that contains single-conformer models of 100 druglike molecules. The conformers and sites will be stored in the database.

1. Start Maestro in the database directory, `phase_tutorial/databases`.

If Maestro is already running, change to the database directory by choosing Change Directory from the Maestro menu in the main window, and navigating to the directory.

2. Choose Applications > Phase > Generate Phase Database in the main window.

The Generate Phase Database panel is displayed.

3.2 Adding Structures to the Database

You can add structures to a database from multiple sources. These sources can contain single molecules, or conformer sets, and both can be stored in the database. When you add structures from a file, you can choose to clean up the structures if they are 2D or are not well optimized.

1. In the Generate Phase Database panel, click Reset at the bottom of the panel to ensure that the default settings are selected.
2. Click Browse to the right of the Input Structure file text box.

A file selector opens.

3. Navigate to the directory `phase_tutorial/databases/userFiles`

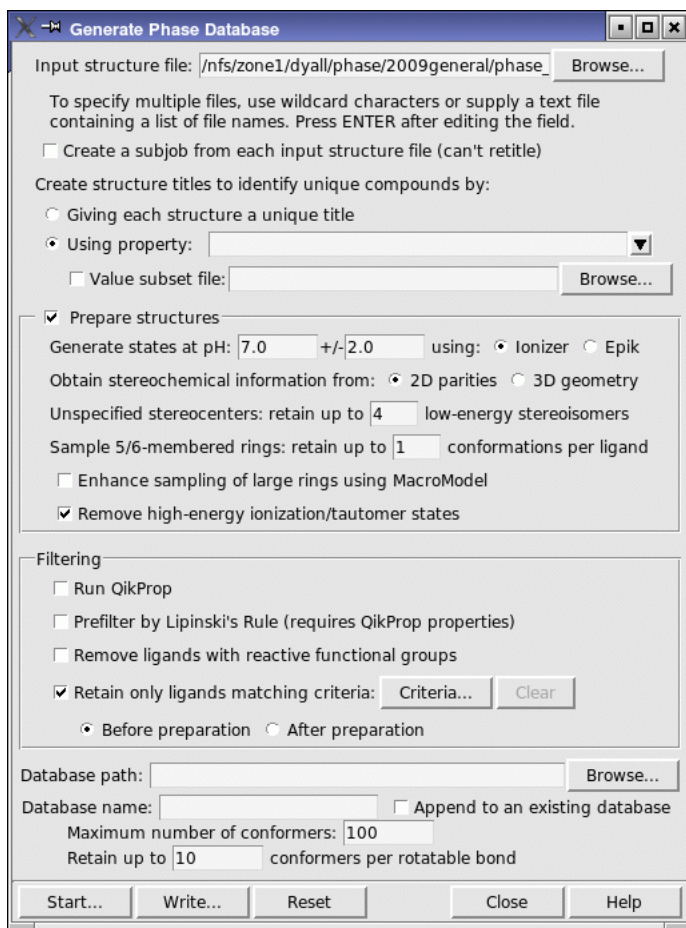


Figure 3.1. The Generate Phase Database dialog box.

4. Select `dbMolecules.mae` and click Open.

The file selector closes, and the file name is displayed in the Input structure file text box..

Each structure added to the database needs a unique title. In this exercise the default, Giving each structure a unique title will be used. However, you may also define structure titles based on selected properties.

5. Ensure that Giving each structure a unique title is selected under Create structure titles to identify unique compounds by.

Databases are often created from chemical files that contain fairly crude structures, so it is important to clean the structures unless you are certain that you have good 3D models with the

proper numbers of hydrogens attached. If you are certain that you have good structures, you can deselect **Prepare structures**. In this exercise you will prepare the structures using the default settings.

6. Ensure that **Prepare structures** is selected.

The structural preparation options should be left at the default settings. You will not be applying any filters so leave the filter options unselected (default).

7. Click **Browse** to the right of **Database path**.
8. Select the **databases** directory.
9. In **Database name** text box type **tutorialDB**.
10. Click **Start** to begin the database generation.

This job takes 15 minutes on a 2 GHz Pentium 4 processor. While the job is running, you can monitor the log files in the **Monitor** panel. Upon completion, the database contains 324 molecules. The number of molecules has increased from 100 to 324 due to the creation of stereoisomers and conformers.

Finding Matches to a Hypothesis

One of the primary reasons for developing a pharmacophore model is to accelerate the identification of new active compounds. This is most commonly done by searching a 3D database for matches to a pharmacophore hypothesis. In this chapter, we demonstrate how an existing pharmacophore hypothesis is used to search the 3D database supplied with the tutorial.

4.1 Preparing for the Exercises

1. Start Maestro in the `phase_tutorial` directory.

If Maestro is already running, change to this directory by choosing Change Directory from the Maestro menu in the main window, and navigating to the directory.

2. Choose Applications > Phase > Find Matches to Hypothesis in the main window.

The Find Matches to Hypothesis panel opens.

Phase database searching is normally broken into two steps: finding and fetching. In the find step, which is the most expensive part of the search, the database is scanned for geometric arrangements of pharmacophore sites that match the hypothesis within a tolerance applied to the intersite distances. A brief summary of each set of matching points is written to a match file, which is subsequently used in the relatively inexpensive fetch step as a lookup table to retrieve hits from the database.

A hit is nothing more than the matching conformation after it has been aligned to the hypothesis using a standard least-squares technique applied to the site points. Hits are returned in order of decreasing fitness, which is a user-adjustable measure of the quality of the alignment of the hit with respect to the reference ligand conformation from which the hypothesis was derived:

$$S = W_{\text{site}} (1 - S_{\text{align}}/C_{\text{align}}) + W_{\text{vec}} S_{\text{vec}} + W_{\text{vol}} S_{\text{vol}}.$$

The terms in the score are described in [Table 4.1](#).

As hits are fetched from the database, they can be filtered using excluded volumes, and their activities can be predicted using a QSAR model. You can also control the maximum total number of hits and the maximum number of hits per molecule.

Table 4.1. Description of parameters in the fitness scoring function.

Parameter	Description
S_{align}	Alignment score: RMS deviation between the site point positions in the matching conformation and the site point positions in the hypothesis.
C_{align}	Alignment cutoff. User-adjustable parameter; default is 1.2.
W_{site}	Weight of site score. User-adjustable parameter; default is 1.0.
S_{vec}	Vector score: average cosine between vector features in the matching conformation and the vector features in the reference conformation.
W_{vec}	Weight of vector score. User-adjustable parameter; default is 1.0
S_{vol}	Volume score: Ratio of the common volume occupied by the matching conformer and the reference conformer, to the total volume (the volume occupied by both). Volumes are computed using van der Waals models of all non-hydrogen atoms.
W_{vol}	Weight of volume score. User-adjustable parameter; default is 1.0

The rationale for separating the steps of finding and fetching is that you might want to explore different settings associated with fitness, excluded volumes, and so on, without having to repeat the more expensive find step. So there are Matching options, which are associated with the find step, and Hit treatment options, which are applied only when fetching.

4.2 Choosing the Database and Hypothesis

Before performing a search, you must select a database and a hypothesis.

1. Click Browse.

A file selector opens, with a filter of `*_phase.inp`.

2. Navigate to the `phase_tutorial/databases` directory.
3. Select the file `testDB_phasedb` and click OK.

The path to the database is now listed in the File name text box.

For the first database search, you will use the subset that contains conformers and sites.

4. Click the Browse button for Subset.
5. In the Select Subset dialog box, select `conformers` and click OK.

The subset name is listed in the Subset text box.

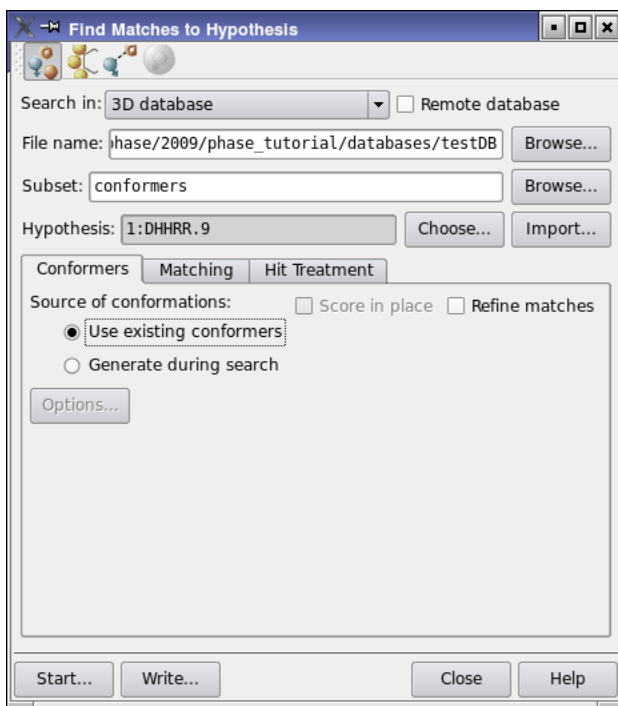


Figure 4.1. The Find Matches to Hypotheses panel.

Next, you will import the hypothesis.

6. Click Import.

A file selector labeled Import Hypothesis opens.

7. Navigate to the directory `phase_tutorial/databases/userFiles`.
8. Select `DHHR.9.xyz` and click OK.

The hypothesis title `DHHR.9` is in the Hypotheses text box, prefixed by the entry ID.

If you click Search for Matches in the Develop Pharmacophore Model panel, the selected hypotheses are added to the Project Table and are available for database searching. Likewise, if you add hypotheses in the Manage Hypotheses panel, these are also available for database searching.

4.3 Performing a Standard Search

In this exercise, you will perform an ordinary search involving both finding and fetching, using default options.

1. In the Conformers tab, ensure that Use existing conformers is selected.
2. In the Hit Treatment tab, ensure that Use QSAR model and Apply excluded volumes are selected.

This ensures that the QSAR models and excluded volumes are used. The remaining options can be left at the defaults.

3. Click Start.

The Start dialog box opens. In this dialog box you can enter a job name, select the host to run the job and the number of processors to use. For the exercises in this chapter, you will run the job on the local host.

4. Enter `conformers` in the Name text box.
5. Click Start in the Start dialog box.

The Monitor panel opens, allowing you to view the progress of the search. You will see output from the find step followed by output from the fetch step. The entire search should take less than a minute, after which a total of 12 hits are placed in the Project Table.

6. Click the Open/Close project table button on the main toolbar.



The Project Table panel is displayed. The hits are in an entry group that is named after the job, `conformers-hits1`. You may view the hits in the Workspace and examine the variety of properties that are written to the Project Table, including the matching sites indices, the fitness and its various components, and the predicted activities from the QSAR model.

4.4 Searching Among Existing Matches

In this exercise you will use the matches found in the previous section, but with different hit treatment, i.e., you will rerun only the fetch step.

1. In the Matching tab, select Use saved matches.
2. In the Hit treatment tab, deselect Apply excluded volumes.

3. In the Hit treatment tab, reduce the hits total setting from 1000 to 5.
4. Click Start, then click Start again in the Start dialog box.
5. When prompted to remove and overwrite job files, click Yes.
6. Click Start.

The Monitor panel shows output from only the fetch step. When the job is finished, a total of five hits appear in the Project Table in an entry group.

4.5 Searching with Site and Conformer Creation

In this exercise you will search the database subset for which conformers and sites were not generated. The conformers and sites will be generated during the search. This kind of search is known as “flexible searching”.

1. In the Find Matches to Hypothesis panel, click the Browse button to the right of the Subset text box.
2. Select the subset `single` and click OK.
3. In the Conformers folder, select Generate during search.
4. In the Matching folder, select Find new matches.
5. In the Hit treatment folder, select Apply excluded volumes and return the hits total setting to 1000.
6. Click Start.
7. In the Name text box of the Start dialog box, enter `single`.
8. Click Start.

The output to the Monitor panel now includes information about the conformation generation. This job should take several minutes, because the conformers are being generated during the search. When the job is finished, a total of 12 hits appear in the Project Table, in an entry group named `single-hits1`. The results should be identical to those obtained in the first search.

Developing Pharmacophore Models from the Command Line

This chapter demonstrates how to develop pharmacophore models and 3D QSAR models by running Phase utilities and modules from the command line.

Phase projects created through the command line are very different from projects created using Maestro, so you will not be able to open a command-line project from Maestro. However, the pharmacophore hypotheses created through the command line are recognized by the Phase GUI, so they can be imported in the Edit Hypotheses and Find Matches to Hypotheses panels.

In order to run the programs described in this tutorial, Phase 3.1 must be installed on the machine you are currently using, as well as on any remote hosts on which jobs will be run. You must also define the `SCHRODINGER` environment variable so that it points to the directory in which the Schrödinger software is installed. To visualize the results of some of the steps, you will also need to be able to run Maestro.

5.1 Pharmacophore Model Utilities

The following utility programs should be present in the directory `$SCHRODINGER/utilities`:

<code>pharm_help</code>	Prints a help message summarizing the command line pharmacophore model workflow, including all the utilities that follow.
<code>pharm_project</code>	Creates a new command line pharmacophore model project and adds molecules to an existing project.
<code>pharm_data</code>	Performs various operations on the project data.
<code>pharm_create_sites</code>	Does setup/cleanup for the job that creates pharmacophore sites.
<code>pharm_find_common</code>	Does setup/cleanup for the job that identifies common pharmacophores.
<code>pharm_score_actives</code>	Does setup/cleanup for the job that scores hypotheses with respect to actives.
<code>pharm_score_inactives</code>	Does setup/cleanup for the job that scores hypotheses with respect to inactives.

<code>pharm_cluster_hypotheses</code>	Does setup/cleanup for the job that clusters hypotheses by geometric similarity.
<code>pharm_build_qsar</code>	Does setup/cleanup for the job that builds QSAR models.
<code>pharm_archive</code>	Preserves project data in a tar archive.
<code>pharm_align_mol</code>	Does setup/cleanup for the job that aligns project ligands or new molecules to a hypothesis.
<code>align_hypoPair</code>	Aligns/merges a pair of hypotheses.
<code>create_xvolShell</code>	Creates a shell of excluded volume spheres around one or more ligands. Provides a means of defining shape-based queries for database searching.
<code>create_xvolClash</code>	Creates excluded volumes using actives and inactives that have been aligned to a hypothesis. Excluded volumes are placed in locations that would cause steric clashes only for the inactives.
<code>create_xvolReceptor</code>	Creates excluded volumes using a receptor structure or a portion thereof.
<code>phase_qsar_stats</code>	Extracts statistics from a hit file that contains QSAR predictions.
<code>qsarVis</code>	Standalone graphical interface for visualizing QSAR models. Available only on Linux-x86 systems.

Except where noted in this tutorial, all changes to project files should be done only through the use of the utilities listed above. Before running any command line utility, it is a good idea to simply type the name of that utility with no arguments to see the usage message.

In addition to the above utilities, you should find the following top-level programs in the `$SCHRODINGER` directory:

<code>phase_feature</code>	Creates pharmacophore sites.
<code>phase_partition</code>	Identifies common pharmacophores.
<code>phase_scoring</code>	Scores hypotheses with respect to actives.
<code>phase_inactive</code>	Scores hypotheses with respect to inactives.
<code>phase_hypoCluster</code>	Clusters hypotheses by geometric similarity.
<code>phase_multiQsar</code>	Builds 3D QSAR models for a collection of hypotheses, and generates a statistical summary for each model.
<code>phase_qsar</code>	Builds a single QSAR model, and generates detailed output.
<code>phase_fileSearch</code>	Aligns structures in a single file to a hypothesis.

5.2 Creating a New Command-Line Project

In this exercise you will create a new project and populate it with a set of 36 endothelin ligands that were taken from the literature (see Krystek, S. R.; Hunt, J. T. Jr.; Stein, P. D.; Stouch, T. R. *Three-Dimensional Quantitative Structure-Activity Relationships of Sulfonamide Endothelin Inhibitors*. J. Med. Chem. **1995**, 38, 659-668).

First, the tutorial files need to be copied from the installation and extracted.

1. Copy `pharm_tutorial.tar.gz` from `$SCHRODINGER/phase-vversion/tutorial` to a directory in which you have write privileges.

2. Extract the file as follows:

```
gunzip -c pharm_tutorial.tar.gz | tar xf -
```

The extraction creates a directory named `pharm_tutorial`.

3. Change to the `pharm_tutorial` directory and create a subdirectory to hold the project:

```
cd pharm_tutorial
mkdir myProj
```

`myProj` will be the root directory for the project, and you must run the pharmacophore model utilities in this directory.

4. Change to the project directory:

```
cd myProj
```

5. Enter the following command to create a new project and import the 36 endothelin ligands:

```
$SCHRODINGER/utilities/pharm_project -new \  
-mae ../userFiles/endo.mae \  
-act r_user_Activity \  
-conf r_mmod_Relative_Potential_Energy-MMFF94s
```

The `-act` option is being used because the Maestro file contains an activity property and we intend to use the activity data to build QSAR models. In the current example, the activity data are `-log[IC50]` values. If the values were not log values, it would be necessary to convert them with `pharm_data`.

If you know the activity values but they are not stored in the Maestro file, then you can manually add them to the file `MasterData.tab` after you import the structures (see Section 1.5). You can omit the `-act` option whenever you create projects in which you do not intend to build a QSAR model or score hypotheses by activity.

The `-conf` option is being used because the Maestro file contains conformers of the ligands with their relative conformational energies, and those values will be used when the hypotheses are scored. If you omit the `-conf` option here, you can still add the conformational energies to the project later using the `pharm_data` utility.

Output is written to the terminal window as each block of conformers is split out from `endo.mae`. When the utility finishes, the following files should be present:

<code>ligands/</code>	Subdirectory that holds all structural data for the project ligands.
<code>ligands/*.mae</code>	Individual ligand files split out from the file <code>endo.mae</code> .
<code>MasterData.tab</code>	A specially formatted text file that holds project data required in various steps of the workflow. Certain modifications are permitted (by hand or through the use of <code>pharm_data</code>).
<code>MasterData.backup</code>	A backup copy of <code>MasterData.tab</code> . Used to revert changes you make to <code>MasterData.tab</code> . Do not modify.
<code>ProjectLigands.inp</code>	Ligand records file. Provides a compact summary of project data, and serves as a template for creating subsets of ligands to align to a hypothesis. While the file can be modified without affecting the integrity of project, it is recommended that you leave it as is, and make a copy of the file if you need to define a subset.
<code>FeatureFreq.tab</code>	Feature frequency file. Used to set minimum and maximum allowed feature frequencies for common pharmacophore perception.
<code>FeatureTol.tab</code>	Feature matching tolerances that can be applied when hypotheses are scored with respect to actives.
<code>pharma_feature.ini</code>	Default pharmacophore feature definitions. You may replace this file with customized definitions, which you would normally create using the Phase panels in Maestro.

5.3 The Master Data File

This section describes the file `MasterData.tab`, which contains various pieces of ligand data that are used throughout the workflow. You may examine this file using a text editor, but be careful not to make any changes right now. In subsequent exercises you will examine this file, so it is useful to have a working knowledge of its contents. If you do not wish to examine this file now, you may proceed to the next section.

At the top of the file, you will find a description of the data it contains, and a number of rules regarding how the data may be modified:

```
#####
#
# Phase Master Data File                                     #
#                                                         #
# You may change PHARM_SET, QSAR_SET, ACTIVITY and 1D_VALUE. To propagate #
# these changes to the project, use 'pharm_data -commit'. To revert to the #
# most recently committed version of the file, use 'pharm_data -restore'. #
#                                                         #
# PHARM_SET: Allowed values are "active", "inactive" and "none". #
#     active - Used to identify common pharmacophores and to score #
#               hypotheses. There must be at least two ligands #
#               with PHARM_SET = active. #
#     inactive - Used to measure the degree to which hypotheses #
#               discriminate actives from inactives. #
#     none - Not used in pharmacophore model development. #
#                                                         #
# QSAR_SET: Allowed values are "train", "test" and "none". #
#     train - Used to develop QSAR models. Recommend at least five #
#             training set ligands for each PLS factor. #
#     test - Used to test QSAR models. #
#     none - QSAR models not applied to these ligands. #
#                                                         #
# ACTIVITY: Ligand activity. Values should increase as potency increases, #
#            for example, -logKi or -logIC50. If activity is unknown, the #
#            value should be "missing". #
#                                                         #
# 1D_VALUE: A conformationally-independent numerical property that may be #
#            used during hypothesis scoring to influence or control the #
#            selection of reference ligands. #
#                                                         #
#####
```

The property corresponding to 1D_VALUE is created for you: it does not come from the original Maestro file you imported. 1D_VALUE allows you to control the set of ligands that can act as a reference when you score hypotheses with respect to actives. By default, any of the actives may be the source of a hypothesis, but if you assign a non-zero 1D_VALUE for certain actives, and a zero value for the remaining actives, then you can force hypotheses to come from only those actives with a non-zero 1D_VALUE. This is similar to using ACTIVITY to bias the selection of reference ligands to favor those with higher activities, but 1D_VALUE allows you to control the selection more precisely.

The Maestro files in the ligands subdirectory contain properties that are linked to those in MasterData.tab by way of the property name block:

```
#####
LIGAND_NAME_PROPERTY = s_phase_Ligand_Name
PHARM_SET_PROPERTY = s_phase_Pharm_Set
QSAR_SET_PROPERTY = s_phase_QSAR_Set
ACT_PROPERTY = r_phase_Ligand_Activity
1D_PROPERTY = r_phase_Ligand_1D_Property
CONF_PROPERTY = r_mmod_Relative_Potential_Energy-MMFF94s
#####
```

Do *not* make any changes to this portion of `MasterData.tab`. The `CONF_PROPERTY` values do not actually appear in the master data file, because they are conformation-dependent, and because there is really no reason you would need to or want to change those values.

Further down the file, you will see a block of data for each ligand stored in the project, like the following:

```
#####
LIGAND_NAME = mol_1
TITLE = "endo-1"
PHARM_SET = active
QSAR_SET = train
ACTIVITY = 5.509
1D_VALUE = 0.0
#####
```

You are allowed to change the values assigned to `PHARM_SET`, `QSAR_SET`, `ACTIVITY`, and `1D_VALUE`. This can be done by hand, or with the aid of the utility `pharm_data`, which allows you to perform a number of numerical and logical operations on the data. This utility is described in detail in [Section 12.3.2](#) of the *Phase User Manual*.

If you make any changes to `MasterData.tab`, whether by hand or through operations supported by `pharm_data`, you must use the `-commit` option to update the Maestro files in the `ligands` subdirectory.

5.4 Defining Active and Inactive Sets

The activity values of the ligands range from 4.276 to 8.398. These are molar pIC_{50} values, so the corresponding concentrations range from about 53 mM to 4 nM. Clearly not all of the molecules in this dataset are highly active, so they cannot all be expected to satisfy the pharmacophore model that will ultimately be generated. The inactives will be used later.

In this exercise you will define an active set and an inactive set, where members of the former are all expected to satisfy the pharmacophore model, whereas one or more members of the latter may not. You will set a threshold of 7.3 for the active set ($\text{IC}_{50} = 50$ nM) and a threshold of 5.0 for the inactive set ($\text{IC}_{50} = 10$ mM).

1. Set the activity thresholds with the following command:

```
$SCHRODINGER/utilities/pharm_data -active 7.3 -inactive 5.0
```

pharm_data reminds you to commit your changes:

```
pharm_data successfully completed
```

Use 'pharm_data -commit' to propagate your changes to the rest of the project

You will be making further changes in the next exercise, so you do not need to commit the changes now.

If you examine the PHARM_SET values in MasterData.tab, you will see that a total of five molecules (mol_5, mol_8, mol_9, mol_16, mol_17) have been assigned to the active set, while a total of six molecules (mol_20, mol_23, mol_29, mol_31, mol_32, mol_34) have been assigned to the inactive set. The remaining molecules fall into a “gray” area of activity and will not be considered when developing pharmacophore models.

5.5 Defining QSAR Training and Test Sets

The project contains a total of 36 molecules, and by default, all of them have been assigned to the QSAR training set. However, you should always define a test set that has no overlap with the training set to validate QSAR models after they are developed, because there is no reason to be confident of a model if it has not been tested on molecules that played no role in developing that model.

In this exercise you will select a random training set of 27 molecules, with the remaining nine molecules going into the test set. In addition, molecules for which PHARM_SET is either active or inactive will not be included in the QSAR test set, but assigned to the training set. Doing so guarantees that the test set has no role in pharmacophore model development, which could ultimately affect the characteristics of the QSAR models. This restriction leads to a test set for which all activities are in the “gray” area as discussed in [Section 5.4](#).

You can include active or inactive molecules in the test set by editing MasterData.tab and changing some of the active and inactive PHARM_SET values to none before choosing the QSAR training and test sets. For this exercise, however, the actives and inactives will be kept in the training set.

1. Enter the following command to randomly assign 27 molecules to the training set, with nine molecules in the test set, subject to the PHARM_SET restriction:

```
$SCHRODINGER/utilities/pharm_data -train 27 -rand 12345678  
-pharm_set -sort
```

The `-sort` option ensures that the training and test sets are sampled uniformly across the range of experimental activities.

2. Examine `MasterData.tab` to verify that there are 9 molecules with a `QSAR_SET` value of test, and that the `PHARM_SET` value for each of those molecules is none.
3. Now commit the changes in `MasterData.tab` to the remainder of the project:

```
$SCHRODINGER/utilities/pharm_data -commit
```

Output is written to the terminal window as each Maestro file in the `ligands` directory is updated. If you now examine the file `ProjectLigands.inp`, you will see a tabular summary of all the committed data, mirroring the property values in `MasterData.tab`.

5.6 Creating Pharmacophore Sites

Before you can develop pharmacophore models, a set of pharmacophore feature definitions must be mapped to the conformations of each ligand. This process identifies pharmacophore sites, which are the geometric locations of hydrogen bond acceptors (A), hydrogen bond donors (D), hydrophobic groups (H), negative ionizable functions (N), positive ionizable functions (P) and aromatic rings (R). Pharmacophore models are developed by enumerating and comparing groups of pharmacophore sites among the ligands in the active `PHARM_SET`.

1. Enter the following command to set up the site creation job:

```
$SCHRODINGER/utilities/pharm_create_sites -setup
```

This command creates the following files:

<code>create_sites_feature.ini</code>	A copy of the default feature definition file <code>pharma_feature.ini</code> .
<code>create_sites_phase.inp</code>	Main input file for <code>phase_feature</code> .

The `create_sites` prefix that appears in each file is linked to the job name that will be specified when the `phase_feature` job is launched. This file naming convention is used throughout the workflow. Because subsequent steps depend on the existence of files created in previous steps, you must not change the names of any input or output job files.

You may examine the `phase_feature` input file, but do not modify it.

2. Start the job by entering the following command:

```
$SCHRODINGER/phase_feature create_sites
```

You can monitor the progress of the job by examining the output written to the file `create_sites_phase.log`.

When the job has finished, you should see the following output at the end of the log file:

```
=====
LIST OF FEATURES
LIGAND NAME : endo-36
=====
feature 0 feature type 0 id A ptype -3 atoms 12 11 13
feature 1 feature type 0 id A ptype -7 atoms 9 7 6
feature 2 feature type 0 id A ptype -7 atoms 10 7 6
feature 3 feature type 0 id A ptype -3 atoms 13 12 14
feature 4 feature type 0 id D ptype -1 atoms 19 8
feature 5 feature type 0 id D ptype -1 atoms 24 23
feature 6 feature type 0 id D ptype -1 atoms 28 23
feature 7 feature type 1 id H ptype 0 atoms 27
feature 8 feature type 2 id H ptype 0 atoms 16
feature 9 feature type 1 id R ptype -8 atoms 11 12 13 14 15
feature 10 feature type 1 id R ptype -8 atoms 1 2 3 4 5 6
=====

write ligand 35 to file ligands/mol_36_sites.phs
conformations 3
total ligands = 36
```

3. Run the cleanup step as follows:

```
$SCHRODINGER/utilities/pharm_create_sites -cleanup
```

This job adds two types of files to the ligands directory:

```
ligands/*_sites.phs Pharmacophore site coordinates for each conformation.
ligands/*_xyz.phc   Atomic coordinates extracted from the ligand Maestro files.
                    These files have a “stripped-down” format that allows rapid
                    access to conformer structural data in subsequent steps of the
                    workflow.
```

Once the cleanup is finished, you will be directed to examine the file `CreateSitesData.tab` for a summary of the results. This file contains the number of occurrences of each type of pharmacophore feature in each project ligand. You may examine this file, but do not make any changes to it, as it is required for the next step in the workflow.

5.7 Finding Common Pharmacophores

Perception of common pharmacophores is carried out with the `phase_partition` module. Briefly, all n -point pharmacophores from the active `PHARM_SET` ligands are enumerated and filtered into a set of high-dimensional boxes. Pharmacophores that fall into the same box are similar enough to be considered equivalent. Boxes that receive at least one pharmacophore from a sufficient number of actives are said to “survive” the partitioning process. For more details, see [Section 5.1](#) of the *Phase User Manual*.

The results of this process depend on the number of sites in the pharmacophore, whether the pharmacophore is required to be common to all actives or a subset of actives, and how many of each kind of feature are permitted or required to be in the pharmacophore. In this exercise, six sites will be used, all five actives must match, and a maximum of three of any kind of feature will be imposed. The pharmacophore minimum and maximum feature frequencies are set in the file `FeatureFreq.tab`.

1. Open `FeatureFreq.tab` in a text editor.

The contents should be as follows:

```
#####
#
# Feature frequency file.  Used to set minimum and maximum allowed feature      #
# frequencies for common pharmacophore perception.  You may change these      #
# limits, but do not make any other modifications to this file.                #
#                                                                              #
#####
A 0 4
D 0 4
H 0 4
N 0 4
P 0 4
Q 0 0
R 0 4
X 0 4
Y 0 4
Z 0 4
END_OF_FEATURE_DATA
```

The lines in the file specify the feature type and the minimum and maximum number of each feature type. For example, the line `A 0 4` indicates that each common pharmacophore will be restricted to contain between zero and four acceptors (inclusive).

The default frequencies should cover every reasonable possibility and even some unlikely ones. In this tutorial you will ensure that the pharmacophore model contains no more than three occurrences of any particular type of feature.

2. Change all of the ranges, except for that of feature type Q, from 0 4 to 0 3.

The lines in the file should be as follows:

```
A 0 3
D 0 3
H 0 3
N 0 3
P 0 3
Q 0 0
R 0 3
X 0 3
Y 0 3
Z 0 3
```

The ranges for Q should not be altered because this feature type has a special meaning that you need not be concerned with at this point. X, Y, and Z are custom feature types, which can be created when you customize your pharmacophore feature definitions. They have no effect on the current project.

3. Save your changes to `FeatureFreq.tab`.
4. Enter the following command to set up the job:

```
$SCHRODINGER/utilities/pharm_find_common -setup -sites 6 -match 5
-freq
```

The `-sites 6` option selects six-point pharmacophores; the `-match 5` option requires that five actives must match the pharmacophore, and the `-freq` option limits the search according to the feature frequencies in `FeatureFreq.tab`. If you omit the `-match` option, it is assumed that you want to match all active set ligands. If you omit the `-freq` option, the default ranges 0 4 will be used for all feature types (except Q).

This command creates the file `find_common_phase.inp`, which is the main input file for `phase_partition`. You do not need to change the settings in this file. For more information on the file format, see [Section B.2](#) of the *Phase User Manual*.

5. Launch the `phase_partition` job with the following command:

```
$SCHRODINGER/phase_partition find_common
```

This job requires about one minute on a 1.7 GHz Pentium 4 machine. You can monitor the progress of the job by examining the file `find_common_partition.log`. When the job is finished, you should see the following output at the end of the log file:

```
PROCESSING VARIANT : 000122
NUMBER OF HYPOTHESES FOR THIS VARIANT : 12
VARIANT DONE !!!
```

THE FOLLOWING VARIANTS WERE REJECTED :

FIND COMMON PHARMACOPHORE STEP COMPLETED SUCCESSFULLY !!!

6. Run the cleanup step as follows:

```
$SCHRODINGER/utilities/pharm_find_common -cleanup
```

You will be directed to examine the file `FindCommonPharmData.tab`, the contents of which should be as follows:

```
#####  
#  
# PHASE Common Pharmacophore Data. NUM_BOXES is the number of surviving boxes #  
# for a given variant, and it represents the maximum number of hypotheses that #  
# can be generated in the Score Actives step. #  
# #  
#####
```

VARIANT	NUM_BOXES
---------	-----------

ADHRRR	65
ADRRRR	58
AADHHR	100
AAHRRR	127
AAADHH	12
AAADRR	80
AAAHRR	118
AADHRR	296
DHRRRR	11
AHRRRR	33
AAADHR	169
AADRRR	24
AAAHHR	59
AAHRRR	46

Total:	1198
--------	------

Each of the 1198 surviving boxes contains a number of very similar pharmacophores, any of which could be considered to be a common pharmacophore. However, using a variety of user-adjustable criteria, we shall select a single pharmacophore from each box, and these will be deemed common pharmacophore hypotheses. The process by which the selections are made is covered in the next section.

5.8 Scoring Hypotheses with Respect to Actives

To discriminate between the various hypotheses, a scoring procedure is used. The first component of the score comes from the geometric alignment of the ligand and the pharmacophore features. Other components can include a term that rewards a high degree of volume overlap, a term for the selectivity of the hypothesis, a reward term for matching a larger number of sites, an activity term and a relative conformational energy term that penalizes high-lying conformations of the ligands. See [Section 2.16 on page 24](#) or [Chapter 6](#) of the *Phase User Manual* for more details about scoring, the underlying algorithms, and the user-adjustable parameters.

1. Set up the scoring job by entering the following command:

```
$SCHRODINGER/utilities/pharm_score_actives -setup -act 1.0
      -conf 0.01
```

Here, the ligand **ACTIVITY** property (as opposed to **1D_VALUE**) is used with a weight of 1.0, and relative conformational energy with a weight of 0.01.

Whenever you incorporate any property into the scoring function, you should examine the range of values to help you arrive at an appropriate weight. The **ACTIVITY** property of the actives spans a range of about 0.5 units, so the chosen weight will lead to a maximum differential of 0.5 in the activity-based scoring term. By contrast, the relative conformational energies for the same ligands are almost two orders of magnitude larger, so a proportionately smaller weight is used to prevent this term from completely dominating the scoring function. The weight is automatically made negative, to ensure that higher energies result in lower scores.

This job creates the main input file `score_actives_phase.inp`. For this exercise, you do not need to change anything in this file.

2. Submit the active scoring job with the following command:

```
$SCHRODINGER/phase_scoring score_actives
```

This job requires about 15 minutes on a 1.7 GHz Pentium 4 machine. You can monitor the progress of the job by examining the file `score_actives_scoring.log`. When the job is finished, you should see the following output at the end of the log file:

```
Scoring Variant AAHRRR ( 14 of 14 )

Starting Score by Vector and Site Alignment ...
Alignment Scoring :   2% done
Alignment Scoring :  10% done
Alignment Scoring :  21% done
Alignment Scoring :  30% done
Alignment Scoring :  41% done
Alignment Scoring :  50% done
```

```
Alignment Scoring : 60% done
Alignment Scoring : 71% done
Alignment Scoring : 80% done
Alignment Scoring : 91% done
Alignment Scoring : 100% done
Starting Score by Volume ...
Volume Scoring : 10% done
Volume Scoring : 20% done
Volume Scoring : 30% done
Volume Scoring : 40% done
Volume Scoring : 50% done
Volume Scoring : 60% done
Volume Scoring : 70% done
Volume Scoring : 80% done
Volume Scoring : 90% done
Volume Scoring : 100% done
Variant Scoring Completed
```

```
*****
```

```
Writing output files. Please, wait ...
Scoring Job Completed Successfully !!!
```

3. Now run the cleanup step as follows:

```
$SCHRODINGER/utilities/pharm_score_actives -cleanup
```

This utility unpacks the archived scoring results and writes summaries to the files `ScoreActivesData.tab` and `ScoreActivesData.csv`. Both files contain the same information, but `ScoreActivesData.tab` is formatted for easy viewing, while `ScoreActivesData.csv` can be loaded into a spreadsheet.

4. Examine the file `ScoreActivesData.tab` with a text editor.

It contains results for 165 hypotheses, an excerpt of which follows:

```
#####
##
#
#          PHASE Hypothesis Scoring Data.  DO NOT MODIFY!!!
#
#####
##
```

HypoID	Survival	Site	Vector	Volume	Prop	Conf	Select	Matches	RefLig
--									
DHHRRR_37	14.9862	0.9662	0.9995	0.8459	8.3980	-0.000	2.7765	5	mol_8
DHHRRR_40	14.8790	0.9643	0.9995	0.8550	8.3980	11.521	2.7775	5	mol_8
DHHRRR_43	14.8454	0.9680	0.9996	0.8253	8.3980	12.212	2.7767	5	mol_8
.									
.									
AADHRR_382	14.2286	0.9651	0.9997	0.8766	8.3980	11.521	2.1044	5	mol_8
AADHRR_372	14.2286	0.9651	0.9997	0.8766	8.3980	11.521	2.1044	5	mol_8

AAAHHR_269	14.2271	0.9692	0.9998	0.8510	8.3980	12.212	2.1313	5	mol_8
.									
.									
AAADRR_22	13.7420	0.9732	0.9997	0.8746	8.3980	11.521	1.6117	5	mol_8
AAADHH_16	13.4861	0.9575	0.9993	0.8745	7.8240	0.286	1.8336	5	mol_16
AAADHH_12	13.4861	0.9575	0.9993	0.8745	7.8240	0.286	1.8336	5	mol_16

For convenience, hypotheses are sorted by decreasing survival score, and there is a column for each quantity that contributes to the survival score. A unique ID is assigned to each hypothesis, according to its variant and the index of its surviving box. The Matches column indicates that all hypotheses matched a total of 5 actives, which is precisely what we required when common pharmacophores were identified. The Site, Vector and Volume scores are all restricted to the interval [0,1], while Selectivity is defined to lie between 0 and 3. A logarithmic scale is used in the estimation of selectivity, so a value of x means that the hypothesis is expected to match only 1 out of every 10^x drug-like molecules by chance. The Prop column is the reference ligand activity and Conf is the reference ligand relative conformational energy.

Observe that mol_8 is the reference ligand for nearly all the hypotheses. This ligand has the highest activity, so incorporation of activity into the scoring function does appear to have yielded the intended result.

For each hypothesis in ScoreActivesData.tab, you will find a series of files in the hypotheses/ subdirectory:

<i>hypoID</i> .def	Feature definitions.
<i>hypoID</i> .mae	Aligned actives.
<i>hypoID</i> .tab	Primary hypothesis data.
<i>hypoID</i> .xyz	Hypothesis site coordinates.

These files have the same format as those created by the Phase GUI when hypotheses are exported, so you can import the primary hypothesis file *hypoID*.tab directly into the Phase GUI (Edit Hypothesis panel or Find Matches to Hypothesis panel). You can also use these hypotheses to search databases through the command line. To set up a command line search, see [Chapter 6](#) for tutorial examples, or [Chapter 13](#) of the *Phase User Manual*.

You may notice that a number of hypotheses from this example are assigned very similar scores. This is not uncommon. Two hypotheses may actually originate from the same reference conformation and belong to the same variant, but differ only in the order in which sites of the same pharmacophoric type are mapped (e.g., $A_1H_2H_3R_4R_5R_6$ vs. $A_1H_3H_2R_4R_5R_6$). So the alignments produced may be very nearly the same, or even identical. In [Section 5.10](#), we demonstrate how a subsequent clustering technique can be applied to identify hypotheses that are essentially equivalent and those that are not.

Even hypotheses that come from different variants may yield very similar scores and alignments. This is fairly common when ligands are built on a common scaffold, because any number of hypotheses may arise from that scaffold. But with information only from actives, it is not always straightforward to determine which of these hypotheses are spurious. Scoring with respect to inactives can provide additional discrimination in such cases.

5.9 Scoring Hypotheses with Respect to Inactives

A molecule that resembles known actives may lack activity itself for any number of reasons, including poor solubility, steric clashes with the receptor, or excessive entropy loss upon binding. These factors are unrelated to the pharmacophore model, but of course there are analogs of actives whose failure to bind is due primarily to a lack of required pharmacophore features. So if a particular hypothesis represents the correct pharmacophore model, then some fraction of inactive molecules would be expected to provide a less than satisfactory match to that hypothesis. This is the premise behind scoring with respect to inactives.

After attempting to obtain alignments for all inactives, an adjusted score is computed:

$$S_{\text{adj}} = S_{\text{actives}} - W_{\text{inactives}} S_{\text{inactives}}$$

where $W_{\text{inactives}}$ is a user-adjustable parameter. Hypotheses that are readily matched by inactives are penalized the most and hence have lowest adjusted scores.

1. Set up the inactive scoring job as follows:

```
$SCHRODINGER/utilities/pharm_score_inactives -setup -w 1.0
```

The `-w` option sets the weight of the inactives score, $W_{\text{inactives}}$.

This command generates the main input file `score_inactives_inactive.inp`. If you do not want to score certain hypotheses you can delete the corresponding `survivalScore` records from this file. For this exercise, leave the file as is.

2. Submit the inactive scoring job as follows:

```
$SCHRODINGER/phase_inactive score_inactives
```

This job requires about two minutes on a 1.7 GHz Pentium 4 machine. When the job is finished, you should see the following output at the end of the log file, `score_inactives_inactive.log`:

```
mol_34: Finding matches to hypothesis AAADHH_16 . . .
mol_34: Number of matches = 20
mol_34: Top fitness value = 2.23454 number of sites matched = 5
mol_34: Finding matches to hypothesis AAADHH_12 . . .
mol_34: Top fitness value = 2.23454 number of sites matched = 5mol_34: Number of matches =
20
```

```
phase_inactive successfully completed
```

3. Now run the cleanup step as follows:

```
$SCHRODINGER/utilities/pharm_score_inactives -cleanup
```

Two summary tables are produced by the cleanup step, `ScoreInactivesData.tab` and `ScoreInactivesData.csv`.

4. Use a text editor to examine `ScoreInactivesData.tab`.

A summary of the contents is as follows:

HypoID	Survival	Inactive	Adjusted

DHRRR_37	14.9862	1.3325	13.6537
DHRRR_40	14.8790	1.7318	13.1472
DHRRR_43	14.8454	1.6474	13.1980
AHRRR_86	14.8453	1.2127	13.6326
DHRRR_42	14.8044	1.6643	13.1401
AHRRR_133	14.7592	1.6885	13.0707
.			
.			
.			
AAARRR_22	13.7420	1.7375	12.0045
AAADHH_16	13.4861	2.5968	10.8893
AAADHH_12	13.4861	2.5968	10.8893

As a result of the comparatively low inactive scores assigned to DHRRR_37 and AHRRR_86, a fairly pronounced gap is now seen between the adjusted scores for these two hypotheses and all other hypotheses. Based on these results, we would now have somewhat greater confidence in DHRRR_37 and AHRRR_86. This doesn't necessarily mean that these hypotheses are the most likely to be correct, but they are at least reasonably consistent with the behavior of the known actives and inactives in the dataset.

5.10 Clustering Hypotheses by Geometric Similarity

As noted in [Section 5.8](#), it is not uncommon to observe two or more hypotheses with very similar or even identical scores and physical characteristics. This is a consequence of the way in which common pharmacophores are perceived (see [Section 5.7](#)). Since the partitioning algorithm operates on an ordered set of intersite distances, it is necessary to consider all permutations among sites of the same type when enumerating pharmacophores. So, for example, the permutations $A_1H_2H_3R_4R_5R_6$ and $A_1H_3H_2R_4R_5R_6$ represent the same basic pharmacophore, but their 15-dimensional intersite distance vectors would generally not be identical, and they may in fact be dissimilar enough to end up in different boxes. As a result, a

given box may have a mirror box that contains many (though not necessarily all) of the same pharmacophores, giving rise to a mirror hypothesis that is indistinguishable (or nearly so) from the original. However, these sorts of redundancies are readily identified, by applying a technique that clusters hypotheses based on geometric similarity.

The top-level program `phase_hypoCluster` performs hierarchical agglomerative clustering of hypotheses based on the following similarity measure applied to a pair of hypotheses i and j , after performing a least-squares alignment of their matching site points:

$$Sim(i, j) = \frac{\langle i | j \rangle}{\sqrt{\langle i | i \rangle \langle j | j \rangle}} \quad (1)$$

where

$$\langle i | j \rangle = alignWeight \cdot Score_{site}(i, j) + vectorWeight \cdot Score_{vector}(i, j)$$

The computation of $\langle i | j \rangle$ is directly analogous to that of $Score_{site+vector+volume}$, except that no volume term is included in $\langle i | j \rangle$, because doing so would incorporate effects that are unrelated to the geometric characteristics of the hypotheses themselves. Thus whereas a volume term may be helpful in determining the best multi-ligand alignment when scoring hypotheses, it is less helpful when computing the geometric similarity between two hypotheses.

In cases where more than one mapping of the site points is possible, the alignment yielding the highest similarity is used. When hypotheses i and j are associated with different variants, the similarity is automatically set to zero. This is done so that the clustering procedure remains true to its original purpose, i.e., distinguishing which hypotheses are geometrically equivalent and which are not. Allowing different variants to cluster together would tend to cloud the situation, when our ultimate goal is to clarify and simplify the hypothesis scoring results. Users who are interested in comparing hypotheses from different variants should use the command line tool `$SCHRODINGER/utilities/align_hypoPair`.

The only option to decide upon for setup is the linkage method, which determines how inter-cluster similarities are measured. This has an impact on the shapes of the clusters that are created, because at each step in the hierarchical, agglomerative clustering process, the most similar pair of clusters is merged. The available linkage methods are:

single	The similarity between clusters is the highest similarity between any two objects from the two clusters. Produces diffuse, elongated clusters.
average	The similarity between clusters is the average similarity between all pairs of objects from the two clusters.
complete	The similarity between clusters is the lowest similarity between any two objects from the two clusters. Produces compact, spherical clusters.

In most cases you will want complete linkage because it tends to yield clusters that are more distinct.

1. Set up the clustering job as follows:

```
$SCHRODINGER/utilities/pharm_cluster_hypotheses -setup
-link complete
```

This command creates the input file `cluster_hypotheses_hypoCluster.inp`. For a description of this file, see [Section B.5](#) of the *Phase User Manual*. For this exercise, leave the file as is.

2. Submit the hypothesis clustering job as follows:

```
$SCHRODINGER/phase_hypoCluster cluster_hypotheses
```

This job requires less than two minutes on a 1.7 GHz Pentium 4 machine. When the job is finished, you should see the following output at the end of the log file:

```
ADHRRR_2      14.6203      +      +
                merge      |
ADHRRR_187    14.6018      +      +
                merge
AADHRR_157    14.0773      +      +
```

```
phase_hypoCluster successfully completed
```

Unlike other jobs, you must examine the contents of the log file to arrive at an appropriate clustering level. The clustering level is specified when the cleanup step is run.

3. Open `cluster_hypotheses_hypoCluster.log` in a text editor.
4. Locate the following section in the file:

		Cluster Counts and Merging Similarities					
		164	163	162	161	160	159
HypoID	Survival	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
AHRRR_109	14.7308	+	+	+	+	+	+
		merge					
AHRRR_113	14.7308	+	+	+	+	+	+
AHRRR_116	14.7054	+	+	+	+	+	+
			merge				
AHRRR_112	14.7054	+	+	+	+	+	+
AHRRR_111	14.6287	+	+	+	+	+	+

This table summarizes the entire clustering process, which starts with 165 singleton clusters, merges the most similar pair to reduce the number of clusters to 164, and then repeats until everything is merged into a single cluster. The vertical order in which the hypotheses appear

reflects the natural groupings established during the clustering process, so when a merge occurs, it always involves two adjacent vertical blocks. Thus, any given cluster always comprises a contiguous range of rows.

In the first step, hypotheses AHRRR_109 and AHRRR_113 are merged into a single cluster. The similarity is 1.0, so these hypotheses are evidently geometrically identical. In the second step, hypotheses AHRRR_116 and AHRRR_112 are merged, also with a similarity of 1.0. The merging of identical geometries continues like this, until the total number of clusters is reduced to 95. This will be apparent if you move down the file until you see the following:

		Cluster Counts and Merging Similarities					
		98	97	96	95	94	93
HypoID	Survival	1.00000	1.00000	1.00000	0.99439	0.99389	0.99372
AHRRR_109	14.7308	+	+	+	+	+	+
AHRRR_113	14.7308	+	+	+	+	+	+
AHRRR_116	14.7054	+	+	+	+	+	+

When the cluster count is reduced to 95, the clusters merged have a similarity of 0.99439. To see where the merge actually occurred, move down the file and look for the merge marker in the corresponding column. You should see the following:

DHRRR_4	13.7734	+	+	+	+	+	+
DHRRR_43	14.8454	+	+	+	+	+	+
					merge		
DHRRR_42	14.8044	+	+	+	+	+	+
DHRRR_41	14.0233	+	+	+	+	+	+

Once again, two individual hypotheses (i.e., singletons) are being merged to form a new cluster.

If you continue to move down the file, you will eventually see:

		Cluster Counts and Merging Similarities					
		92	91	90	89	88	87
HypoID	Survival	0.99358	0.99335	0.98817	0.98768	0.98306	0.98104
AHRRR_109	14.7308	+	+	+	+	+	+
AHRRR_113	14.7308	+	+	+	+	+	+
AHRRR_116	14.7054	+	+	+	+	+	+
AHRRR_112	14.7054	+	+	+	+	+	+
		merge					
AHRRR_111	14.6287	+	+	+	+	+	+

AHRRR_115	14.6287						
		+	+	+	+	+	+
AHRRR_136	14.7314	+	+	+	+	+	+

Here you see a pair of non-singleton clusters being joined. The merging similarity is 0.99358, which corresponds to the most dissimilar pair of hypotheses between the two clusters, because complete linkage is being used. It's important to keep in mind that the merge similarity is not necessarily the similarity between the two vertically adjacent hypotheses (in this case AHRRR_112 and AHRRR_111), but rather the similarity between the two vertically adjacent clusters, and that similarity depends on the linkage method.

At this stage, the clusters being merged are still highly similar, so the hypotheses contained in them are probably almost indistinguishable. What we are interested in is identifying the point at which a fairly pronounced drop occurs in the merging similarity, which indicates that clusters are being joined that contain hypotheses with more noticeable differences. If you move further down the file you should see:

		Cluster Counts and Merging Similarities					
		74	73	72	71	70	69
HypoID	Survival	0.97594	0.97585	0.97583	0.97529	0.97192	0.86978
AHRRR_109	14.7308	+	+	+	+	+	+
AHRRR_113	14.7308	+	+	+	+	+	+
AHRRR_116	14.7054	+	+	+	+	+	+

So, in forming 69 clusters, the merging similarity drops from 0.97192 to 0.86978, which is by far the most dramatic drop seen so far. However, there is an even more pronounced drop in forming 68 clusters:

		Cluster Counts and Merging Similarities					
		68	67	66	65	64	63
HypoID	Survival	0.73068	0.71944	0.71658	0.70925	0.70832	0.70726
AHRRR_109	14.7308	+	+	+	+	+	+
AHRRR_113	14.7308	+	+	+	+	+	+
AHRRR_116	14.7054	+	+	+	+	+	+

So we can be fairly confident that the original 165 hypotheses are reasonably well represented by 70 or so clusters. With this knowledge, you can run the cleanup step and request an easily interpreted report for the formation of 70 clusters.

5. Run the cleanup step as follows:

```
$SCHRODINGER/utilities/pharm_cluster_hypotheses -cleanup
-report 70
```

You can run the cleanup step as many times as you wish in order to examine reports for different clustering levels. When the utility has finished, you are directed to examine the file `ClusterHypothesesData.tab`, whose contents are summarized as follows:

HypoID	Survival	Cluster	Size	AvgSim

AHHRRR_109	14.7308	1	6	0.98398
AHHRRR_113	14.7308	1	6	0.98398
AHHRRR_116	14.7054	1	6	0.98812
AHHRRR_112	14.7054	1	6	0.98812
AHHRRR_111	14.6287	1	6	0.99073
AHHRRR_115	14.6287	1	6	0.99073

AHHRRR_136	14.7314	2	3	0.98584
AHHRRR_135	14.6595	2	3	0.98895
AHHRRR_133	14.7592	2	3	0.98091

AHHRRR_86	14.8453	3	1	0.00000

.				
.				
.				

AADHRR_317	14.1870	65	2	0.97585
AADHRR_318	14.1591	65	2	0.97585

DHHRRR_37	14.9862	66	1	0.00000

ADHRRR_117	14.7499	67	1	0.00000

ADHRRR_2	14.6203	68	1	0.00000

ADHRRR_187	14.6018	69	1	0.00000

AADHRR_157	14.0773	70	1	0.00000

From this clustering report you can readily identify instances where a large number of hypotheses are satisfactorily represented by a single hypothesis. In selecting a representative of a given cluster, you may wish to choose the hypothesis with the highest survival score, or the one that exhibits the highest average similarity to other members of its cluster, i.e., the hypothesis that best represents the cluster in an average sense.

To see exactly how similar the clustered hypotheses are, the two most dissimilar hypotheses in any single cluster can be aligned with the `align_hypoPair` utility. These two hypotheses are `AAADHH_11` and `AAADHH_12`, with a similarity of 0.97192.

- Run the align_hypoPair utility as follows:

```
$SCHRODINGER/utilities/align_hypoPair -fixed hypotheses/AAADHH_11
    -free hypotheses/AAADHH_12 -new NEW
```

The utility finds two 6-point mappings, and by default creates a new hypothesis (with file prefix NEW), for the best one. You should see the following summary for the top-ranked match:

Match 1: RMSD = 0.0663, Number of sites matched = 6

Fixed		Free		New		Deviation
1 A	<-->	1 A	1 A	3.0520	-2.1851	-1.4923
2 A	<-->	2 A	2 A	5.2082	-0.9840	-1.9395
3 A	<-->	0 A	0 A	6.2054	0.1666	0.7228
4 D	<-->	4 D	4 D	4.1863	-2.4586	0.9373
6 H	<-->	5 H	5 H	5.4968	-2.3093	3.3994
7 H	<-->	6 H	6 H	7.6892	0.1042	4.0108
						0.0552
						0.0792
						0.0647
						0.0901
						0.0433
						0.0535

Thus when AAADHH_12 is aligned onto AAADHH_11, the deviations in the site point positions are all less than 0.1 Å, which is quite tiny compared to the tolerances normally used when searching a database for matches to a hypothesis. Consequently, essentially every molecule that matches AAADHH_11 would also match AAADHH_12.

5.11 Building 3D QSAR Models

While scoring with respect to inactives provides an indication of the ability of hypotheses to discriminate between molecules with high and low activities, QSAR models tackle the problem across a continuous range of activities. In this section, we develop 3D QSAR models for all hypotheses at once, and we demonstrate how the most interesting models can be examined in detail. Phase QSAR models are generated by partial least-squares fitting of a large number of binary values that represent the occupancy of cubic regions of space by the atoms or pharmacophores in the ligands. For details, see [Chapter 7](#) or [Appendix A](#) of the *Phase User Manual*.

You can select the model type (atom or pharmacophore), the grid spacing, and the maximum number of PLS factors in the model. In this exercise, you will use the defaults for the model type (atom) and the grid spacing (1.0 Å), and a maximum of three PLS factors.

- Set up the job with the following command:

```
$SCHRODINGER/utilities/pharm_build_qsar -setup -factors 3
    -tvalue 2.0 -exclude 3 -rand 123456789
```

This command creates the main input file build_qsar_multiQsar.inp. You do not need to modify this file.

The option `-exclude 3` indicates that we want to estimate *t*-values by holding out random sets of 3 compounds from the training set, and `-rand 123456789` is the random seed for this process.

2. Submit the multiple QSAR job as follows:

```
$SCHRODINGER/phase_multiQsar build_qsar
```

This job requires about 15 minutes on a 1.7 GHz Pentium 4 processor. You may monitor the job's progress by examining the file `build_qsar_multiQsar.log`. When the job is finished, you should see output like the following at the end of the log file:

```
Building QSAR model for AAADRR_22 . . .
Building QSAR model for AAADHH_16 . . .
Building QSAR model for AAADHH_12 . . .

Creating resultArchive build_qsar_results.tar . . .

CPU time = 798.07 sec

phase_multiQsar successfully completed

Driver script for parent phase_multiQsar job build_qsar finished
Current time: Tue May 8 22:04:04 2007
Elapsed time = 00:14:00

phase_multiQsar results for build_qsar are complete2
```

3. Now run the cleanup step:

```
$SCHRODINGER/utilities/pharm_build_qsar -cleanup
```

4. Use a text editor to examine the file `BuildQsarData.tab`.

You will see a number of statistical quantities (see [Appendix A](#) of the *Phase User Manual* for further details):

SD	Standard deviation of regression. An effective RMS error in the training set fit, corrected for the degrees of freedom in the model.
R ²	Coefficient of determination. Measures the fraction of the variance in the training set activity data that's accounted for by the model.
F	Variance ratio statistic for the model. Larger values indicate greater statistical significance.
P	Significance level of F. Smaller values indicate greater confidence.

RMSE	RMS error in the test set predictions.
Q ²	Analogous to R ² , but computed with respect to the test set predictions. Can be negative if the average squared error is larger than the variance in the test set activity data.
R-Pearson	Pearson correlation coefficient computed between the predicted and observed test set activities. Ranges from -1 to 1.

Ideally, a model should yield a large F value and comparable prediction errors for both the training and test sets, i.e., $SD \approx RMSE$. Comparing R^2 and Q^2 is not as meaningful since the test set may not span as large a range of activities as the training set, leading to small (and sometimes negative) Q^2 values, even if the predictions are relatively accurate. Moreover, the test set activities may be accurately predicted in a relative sense, but a slight systematic error can drastically reduce Q^2 . R-Pearson provides a better measure of relative accuracy.

It is important to recognize that there is no single statistic that unequivocally determines which model is best. The battery of statistics should be considered in totality, and some measure of common sense must be applied. For example, IC_{50} values may only be accurate to a multiplicative factor of 2, so the corresponding $-\log[IC_{50}]$ values would only be accurate to $\log(2)$. So if the SD statistic is smaller than this experimental uncertainty, then the data are clearly being over-fit, and the model is bound to yield spurious predictions on certain molecules outside the training set, even if the test set predictions appear satisfactory.

To illustrate how you might decide which models are superior, consider the results for DHRRR_37. When three PLS factors are used, there is a very tight fit of the training set data, and the high F value indicates a high level of statistical significance compared to most other models. However, the test set RMSE is 50% larger than the training set SD, so the model does not appear to have a great deal of absolute predictive power, at least with respect to the current test set. Further, the Pearson-R value is around 0.6, so the relative predictive power is not that impressive either.

By contrast, consider the 3-factor model for ADHRR_57. Here, the training set is well fit, the statistical significance is high (large F), and the test set RMSE is comparable to the training set SDSD, and Q^2 and Pearson-R indicate satisfactory accuracy in both absolute and relative senses. Accordingly, we would view this model as superior to the model for DHRRR_37.

The summary provided in `BuildQsarData.tab` is very helpful for identifying the most promising models. But no doubt you will want more information about certain models. In the subdirectory `BuildQsarResults` you will find a number of files for each hypothesis:

<i>hypoID_align.mae</i>	Aligned structures for training and test set molecules. Training set molecules appear first.
<i>hypoID.def</i>	Feature definitions.
<i>hypoID.mae</i>	Reference ligand structure.
<i>hypoID_order.dat</i>	File that defines the overall order of molecules in <i>hypoID_align.mae</i> .
<i>hypoID.qsar</i>	QSAR model file.
<i>hypoID_qsar.inp</i>	Main input file for a <i>phase_qsar</i> job.
<i>hypoID.tab</i>	Primary hypothesis data, with QSAR model flag activated.
<i>hypoID.xyz</i>	Hypothesis site coordinates.

The Phase GUI recognizes the primary hypothesis file *hypoID.tab* and database searches conducted with any of these hypotheses automatically apply the associated QSAR model to the hits.

More importantly, this directory contains everything you need in order to run a *phase_qsar* job for any particular hypothesis. *phase_qsar* writes out complete details of the model, along with predictions for the training and test set molecules.

5. Change to the *BuildQsarResults* directory:

```
cd BuildQsarResults
```

6. Use a text editor to examine the input file *DHRRR_1_qsar.inp*.

You will see a number of options, which you need not change, followed by information about the indexing of ligands. The training set molecules are placed before the test set molecules, and you may wish to refer to these indices when you examine out output from *phase_qsar*.

7. Submit the *phase_qsar* job with the following command:

```
$SCHRODINGER/phase_qsar DHRRR_1
```

This job should take only 2-3 seconds. You may verify that it has finished by examining the file *DHRRR_1_qsar.log*, which should contain only the message:

```
phase_qsar successfully completed
```

8. Use a text editor to examine the file *DHRRR_1_qsar.out*.

You will find complete model statistics, the Cartesian coordinates and regression coefficient for each “bit” in the model, training and test set predictions, and the actual bit values for each molecule. These bit values could in fact be extracted from the file and used to

build an equivalent QSAR model outside the framework of Phase. They could also be combined with other independent variables (such as logP) to build models that incorporate additional effects. The training and test set predictions can be extracted and copied to text files, which can be imported into a spreadsheet program to create a scatter plot.

9. Before proceeding to the next section, return to the root project directory:

```
cd ..
```

5.12 Visualizing QSAR Models

Phase includes a utility, `qsarVis`, that can display a visual representation of QSAR models that were developed from the command line. The display has the same appearance as the corresponding Maestro display—see [Section 2.22 on page 38](#).

This utility only runs under Linux. If you are not running under Linux, skip to the next exercise.

To run this utility, you must supply the name of the hypothesis used to create the QSAR model, and the name of a single molecule in the project. By default, you will be viewing the bits set for that molecule, for which the associated regression coefficients exceed applicable positive and negative thresholds. This allows you to identify favorable (blue) and unfavorable (red) regions of the molecule, and to control whether you see weak, moderate or strong effects (larger threshold → stronger effect on activity).

First you will visualize the 3-factor QSAR model for hypothesis `ADHRRR_57`, displaying the bits set for its reference ligand, `mol_8`, which is the most active molecule in the data set.

1. Enter the following command:

```
$SCHRODINGER/utilities/qsarVis -hyp BuildQsarResults/DHRRR_1 \  
-mol mol_8 -pc 0.025 -nc -0.025 -npls 3 &
```

The `-pc` and `-nc` options specify the regression coefficient tolerances. These are about midway between zero and the largest coefficient magnitude. The `-mol` option specifies the molecule, and the `-npls` option specifies the number of PLS factors in the model. The command is run in the background so that you can have more than one view available at a time.

This command starts a graphical interface entitled *Visualization Toolkit – OpenGL*, with an interactive 3D image of the ligand, hypothesis, and QSAR model. To rotate the image, drag with the left mouse button; to translate, drag with the middle mouse button; to zoom, drag with the right mouse button.

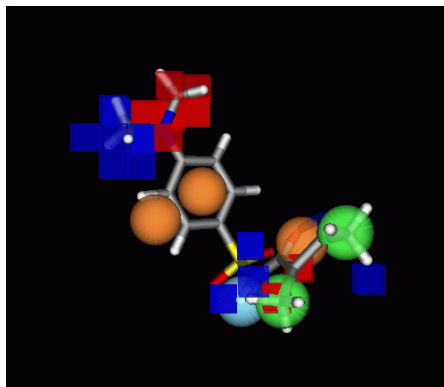
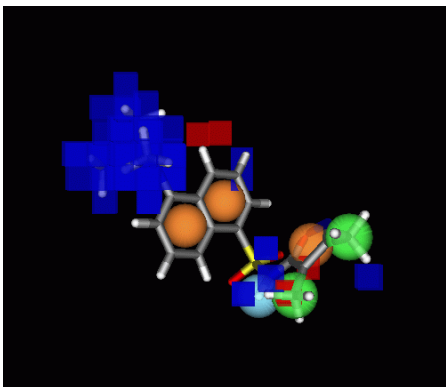


Figure 5.1. View of the DHRRR_1 QSAR model for the strongly active ligand mol_8 (left) and the inactive ligand mol_20 (right). Effects from all atom classes are represented.

The blue regions of the molecule imply favorable interactions with the receptor, whereas the red regions imply unfavorable interactions. As expected, favorable interactions predominate for this highly active molecule. Figure 5.1 provides a view for this ligand.

As a contrast, you can visualize the bits set for the least active molecule as follows:

2. Enter the following command:

```
$SCHRODINGER/utilities/qsarVis -hyp BuildQsarResults/DHRRR_1  
-mol mol_20 -pc 0.025 -nc -0.025 -npls 3 &
```

The unfavorable interactions are now far more prevalent than they were for mol_8, as shown in Figure 5.1.

So far, you have been visualizing the combined effects of all atom types, but you can restrict the view to show only the bits set by a particular class of atom.

Enter the following command to view only the effects attributed to electron-withdrawing atoms:

```
$SCHRODINGER/utilities/qsarVis -hyp BuildQsarResults/DHRRR_57  
-mol mol_8 -class W -pc 0.01 -nc -0.01 -npls 3 &
```

The `-class` option selects the atom class. The regression coefficient thresholds have been lowered so that even weak effects will be visible. The occupied volumes are now restricted to regions around oxygens and nitrogens, as shown in Figure 5.2.

3. Close the visualization windows.

You can ignore the error messages.

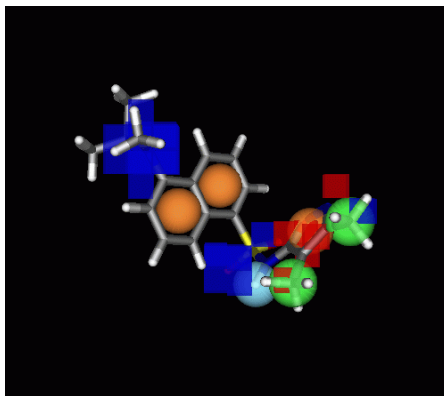


Figure 5.2. View of the DHRRR_1 QSAR model for the strongly active ligand mol_8. Only the effects from electron-withdrawing atoms are represented.

5.13 Applying Models to New Molecules

While hypotheses and QSAR models can be used to align or make predictions for molecules that are stored in a 3D database, you may not want to go to the trouble of creating a database if you simply want to apply a model to a relatively small number of molecules that you didn't include in the original project. This section demonstrates how to perform a file-based search on a set of new molecules, which may or may not be represented by multiple conformers.

File-based searches are set up by running `pharm_align_mol`. You can apply models either to project ligands or to molecules stored in an external Maestro or SD file. In this section you will use an external file. Alignment of project ligands is covered in [Section 5.14](#), in conjunction with the creation of excluded volumes.

5.13.1 Standard Search

In this exercise, each of the molecules in the file to be searched is already represented by multiple low-energy conformers.

1. Enter the following command to set up the file-based searching job:

```
$SCHRODINGER/utilities/pharm_align_mol -setup align_confs  
-hypo BuildQsarResults/DHRRR_1 -mol ../userFiles/endo.mae  
-minSites 3 -phase build_qsar_phase.inp
```

The file is the Maestro file used to create the project initially, but it could be any multi-conformer Maestro file. The `-phase` option is not necessary, but it guarantees that certain matching parameters are identical to those used when the QSAR model was built. If you want to search using one of the hypotheses in the `hypotheses` directory that does

not have a QSAR model, you could supply `-phase score_hypotheses_phase.inp`. If you wish to accept the default matching parameters, you can omit the `-phase` option.

2. Submit the file-based searching job as follows:

```
$SCHRODINGER/phase_fileSearch align_confs
```

This job requires only a few seconds on a 1.7 GHz Pentium 4 machine. You may monitor the progress of the job by examining the file `align_confs_fileSearch.log`. When the job is finished, you should see the following output at the end of the log file:

```
Processing endo-36 . . .
Number of conformations read = 3
Found 4 matches . . .
Writing 1 hit to align_confs-hits.maegz
Total hits written so far = 36

File-based search finished
Total number of molecules processed = 36
Total number of hits written to align_confs-hits.maegz = 36
CPU time = 6.47 sec

phase_fileSearch successfully completed
```

3. Run the cleanup step as follows:

```
$SCHRODINGER/utilities/pharm_align_mol -cleanup align_confs
```

To view the aligned hits, you can import the file `align_confs-hits.maegz` into Maestro. The properties created during the search, including predictions from the QSAR model, are displayed in the Project Table. You can create a scatter plot of predicted activity against experimental activity from the data in the Project Table, using the Plot panel.

The utility `phase_qsar_stats` gives a summary of the QSAR predictions in the hit file just created, and can be used to create a comma-separated value (CSV) file that can be imported into a spreadsheet program to create a scatter plot.

4. Enter the following command to obtain QSAR statistics:

```
$SCHRODINGER/utilities/phase_qsar_stats
-hypo BuildQsarResults/DHRRR_1 -hits align_confs-hits.maegz
-act r_user_Activity -plot align_confs.csv
```

The following output should be written to the terminal window:

```
Model Summary
-----

Model Type = "atom"
Number of PLS Factors = 3
```

Grid Spacing = 1.00

Grid Limits:

	Min	Max
x	-11.000	16.000
y	-11.000	14.000
z	-9.000	12.000

PLS Regression Statistics:

#Factors	S.D.	R-Squared	F	P
1	0.6707	0.7308	67.9	1.368e-08
2	0.5001	0.8563	71.5	7.726e-11
3	0.4166	0.9045	72.6	7.067e-12

PLS Regression Coefficients and T-Values:

Factor	Coeff	T(#Factors in Regression)		
		T(1)	T(2)	T(3)
1	0.2856	8.239	11.050	13.264
2	0.1756		4.579	5.496
3	0.1121			3.403

PLS Regression Coefficient Limits

Class	#Factors	Min	Max
D	1	-2.182e-02	2.015e-02
	2	-1.911e-02	2.928e-02
	3	-2.372e-02	5.378e-02
H	1	-2.571e-02	3.187e-02
	2	-4.366e-02	4.761e-02
	3	-6.786e-02	5.368e-02
W	1	-2.849e-02	3.189e-02
	2	-2.581e-02	4.791e-02
	3	-2.900e-02	5.356e-02
X	1	-2.015e-02	2.015e-02
	2	-1.283e-02	1.821e-02
	3	-1.439e-02	2.524e-02
All	1	-3.127e-02	4.849e-02
	2	-3.884e-02	8.310e-02
	3	-5.147e-02	8.848e-02

Activity Predictions

Number of molecules with experimental activities = 36
 Number of molecules predicted = 36

#Factors	Variance(Exp)	Variance(Pred)	RMSE	Q ²	R-Pearson
1	1.2673	0.9689	0.6344	0.6825	0.8282
2	1.2673	1.0832	0.4557	0.8362	0.9150
3	1.2673	1.1295	0.3883	0.8810	0.9389

Activity_Exp	Activity_Pred(1)	Activity_Pred(2)	Activity_Pred(3)
5.5090	6.4337	6.0821	5.9150
5.4560	6.2452	5.8540	5.8190
5.4690	6.3468	5.7563	5.4305
.			
.			
.			
4.2760	5.3558	5.0652	5.3673
6.5850	5.6351	5.5981	6.2113
6.2920	5.5558	5.5872	6.2392

You can use this file in a spreadsheet to generate a scatter plot of predicted activity against experimental activity, for example.

5.13.2 Flexible Search

If you want to search a file that contains only one conformer per molecule, you can request a flexible search, where conformers are generated as needed. Although these conformers are not fully minimized, they are satisfactory for identifying whether or not a molecule can match a hypothesis. If reliable activity predictions are sought, however, it is recommended that fully minimized conformers be created using MacroModel, in a manner consistent with the original conformers from which the pharmacophore and QSAR models were developed.

This exercise demonstrates flexible file-based searching using the same 36 endothelin ligands as in the last section, but starting with only a single low-energy structure for each molecule.

1. Set up the flexible search with the following command:

```
$SCHRODINGER/utilities/pharm_align_mol -setup align_flex \
  -hypo BuildQsarResults/DHRRR_1 \
  -mol ../userFiles/endo_1conf.mae -flexMaxConfs 100 \
  -minSites 6 -phase build_qsar_phase.inp
```

The `-flexMaxConfs 100` option specifies that a maximum of 100 conformers per molecule will be generated in the flexible search. The `-match 6` option requires that all six sites in the hypothesis must be matched.

2. Submit the `phase_fileSearch` job as follows:

```
$SCHRODINGER/phase_fileSearch align_flex
```

This job also requires only a few seconds on a 1.7 GHz Pentium 4 machine. When the job is finished, you should see the following output at the end of the log file:

```
Processing endo-36 . . .
Number of conformations read = 1
** Entering Conformation Generator **

Number of rotatable bonds          3
Core rotatable bonds, maxkeep      3      1000
Maximum excitation level          11
Energy cutoff (kcal/mole)         10.000
Total core conformations           4
Total conformations minimized      4

Number of flexible conformations stored = 4
Found 0 matches . . .

File-based search finished
Total number of molecules processed = 36
Total number of hits written to align_flex-hits.mae = 27
CPU time = 14.67 sec

phase_fileSearch successfully completed
```

Of the 36 molecules, 9 failed to produce any 6-point matches. These are precisely the same molecules that yielded only 5-point and 4-point matches in the previous search when fully minimized conformers were used.

3. Run the cleanup step to remove unnecessary intermediate files:

```
$SCHRODINGER/utilities/pharm_align_mol -cleanup align_flex
```

5.14 Creating Excluded Volumes

A molecule may satisfy a pharmacophore model, yet fail to bind to the associated receptor due to steric clashes. Excluded volumes provide a means of restricting the space that can be occupied by any molecule that matches a pharmacophore model, thereby incorporating steric constraints imposed by the receptor. There are three utilities that allow you to create excluded volumes in an automated fashion, using varying amounts of ligand and receptor information:

<code>create_xvolShell</code>	Creates a shell of excluded volume spheres around one or more ligands. Provides a means of defining shape-based queries for database searching.
<code>create_xvolClash</code>	Creates excluded volumes using actives and inactives that have been aligned to a hypothesis. Excluded volumes are placed in locations that would cause steric clashes only for the inactives.
<code>create_xvolReceptor</code>	Creates excluded volumes using a receptor structure or a portion thereof.

5.14.1 Creating a Shell Around a Ligand

The first exercise uses the utility `create_xvolShell`. You would normally run this utility if you believe that one or more ligands effectively sweep out the space of the binding pocket. In other words, if a molecule that is aligned to the hypothesis cannot fit into the shape defined by the excluded volume shell, then it is presumed that the molecule would experience a steric clash with the receptor.

Because the shell normally obscures the view of the ligands it surrounds, it is a good idea to start off with the `-cut` option to create only half the shell. This allows you to verify that the shell is in fact being created the way you expected, after which you can create the full shell.

1. Enter the following command to create half a shell around the reference ligand from the ADHRRR_57 hypothesis:

```
$SCHRODINGER/utilities/create_xvolShell \  
-hypo BuildQsarResults/DHRRR_1 -buff 2.0 -cut
```

A message that the excluded volumes file `BuildQsarResults/ADHRRR_57.xvol` has been created is displayed. To view the excluded volumes, simply import this hypothesis into the Edit Hypothesis panel in Maestro and click the Excluded Volumes toolbar button.

If you choose the CPK molecular representation for the ligand with CPK percentage equal to 100, you will see that the surfaces of all heavy atoms in the ligand are at least 2 Å from the surface of any excluded volume sphere.

2. Now go back and create the full excluded volume shell:

```
$SCHRODINGER/utilities/create_xvolShell \  
-hypo BuildQsarResults/DHRRR_57 -buff 2.0
```

This hypothesis could now be used to search a database or a structure file for matches to the hypothesis, with the restriction that the aligned structures fit within the shell. If you wish to exercise additional control over the shape of each match, you can search using the `volumeCutoff` option (see [Section 13.3](#) and [Section B.13](#) of the *Phase User Manual*) to elim-

inate matches that fit within the shell, but which you would consider to be too small relative to the reference ligand. For example, setting `volumeCutoff` equal to 0.9 would force all matches to overlap 90% with the reference ligand.

5.14.2 Using Inactives to Define Steric Clash Regions

In this exercise you will create excluded volumes using information from both actives and inactives. The utility `create_xvolClash` automatically places excluded volume spheres in positions that would create clashes only for the inactives. Before you can run the `create_xvolClash` utility, you must align the actives and inactives to a hypothesis, using `pharm_align_mol` and `phase_fileSearch`. To do this, you must create files that contain `LIGAND_NAME` records for just those molecules.

1. Copy the input files as follows:

```
cp ProjectLigands.inp actives.inp
cp ProjectLigands.inp inactives.inp
```

2. Use a text editor to open the file `actives.inp` and delete all `LIGAND_NAME` records for which `PHARM_SET` is not active.

The contents of the file should now be as follows:

```
#####
#
#   Ligand Records File.  All text following a pound sign "#" is ignored.
#
#####
LIGAND_DIR = ligands      #   PHARM_SET   QSAR_SET   ACTIVITY   1D_VALUE   #
#####
LIGAND_NAME = mol_5      #   active     train      7.824      0.0        #
LIGAND_NAME = mol_8      #   active     train      8.398      0.0        #
LIGAND_NAME = mol_9      #   active     train      8.301      0.0        #
LIGAND_NAME = mol_16     #   active     train      7.824      0.0        #
LIGAND_NAME = mol_17     #   active     train      7.796      0.0        #
```

3. Edit the file `inactives.inp` and delete all `LIGAND_NAME` records for which `PHARM_SET` is not inactive.

The contents of the file should now be as follows:

```
#####
#
#   Ligand Records File.  All text following a pound sign "#" is ignored.
#
#####
LIGAND_DIR = ligands      #   PHARM_SET   QSAR_SET   ACTIVITY   1D_VALUE   #
#####
LIGAND_NAME = mol_20      #   inactive    train      4.076      0.0        #
```

```
LIGAND_NAME = mol_23      #   inactive   train    4.62      0.0      #
LIGAND_NAME = mol_29      #   inactive   train    4.377     0.0      #
LIGAND_NAME = mol_31      #   inactive   train    4.502     0.0      #
LIGAND_NAME = mol_32      #   inactive   train    4.561     0.0      #
LIGAND_NAME = mol_34      #   inactive   train    4.276     0.0      #
```

Before proceeding, however, you should consider whether or not there are any obvious inconsistencies in assuming that all of these molecules are inactive because of steric clashes. You will be creating excluded volumes for hypothesis DHRRR_1, and according to the file-based search results from [Section 5.13](#), mol_34 matches only 5 out of 6 sites in this hypothesis. So there is some question as to whether mol_34 is inactive because of steric clashes or because it is missing a critical interaction with the receptor. But note that mol_33 matches the same 5 out of 6 sites as mol_34, with almost identical fitness, yet its activity is 5.398, a full order of magnitude higher. Hence it would appear that additional factors are contributing to the inactivity of mol_34, so we shall retain it for purposes of creating excluded volumes.

Since you have already created excluded volumes for the DHRRR_1 hypothesis, you should remove the associated file before running `pharm_align_mol`:

4. Remove the file `DHRRR_1.xvol`:

```
rm BuildQsarResults/DHRRR_1.xvol
```

This prevents those excluded volumes from being applied when we create alignments for the actives and inactives. You could also move this file to some other location if you wanted to keep the excluded volumes.

5. Set up the job to align actives as follows:

```
$SCHRODINGER/utilities/pharm_align_mol -setup align_actives
-hypo BuildQsarResults/DHRRR_1 -lig actives.inp
-minSites 6 -phase build_qsar_phase.inp
```

6. Submit the `phase_fileSearch` job as follows:

```
$SCHRODINGER/phase_fileSearch align_actives
```

7. When the job finishes, run the cleanup step:

```
$SCHRODINGER/utilities/pharm_align_mol -cleanup align_actives
```

8. Now set up the job to align inactives:

```
$SCHRODINGER/utilities/pharm_align_mol -setup align_inactives
-hypo BuildQsarResults/DHRRR_1 -lig inactives.inp
-minSites 3 -phase build_qsar_phase.inp
```

In this case, we are requiring that only 3 of 6 sites be matched, although we know from previous work that `-match 5` would achieve the same results.

9. Submit the `phase_fileSearch` job as follows:

```
$SCHRODINGER/phase_fileSearch align_inactives
```

10. When the job finishes, run the cleanup step:

```
$SCHRODINGER/utilities/pharm_align_mol -cleanup align_inactives
```

At this point you have two hit files, which you can use as input to `create_xvolClash`.

11. Enter the following command to create excluded volumes that will clash only with the inactives:

```
$SCHRODINGER/utilities/create_xvolClash  
-hypo BuildQsarResults/DHRRR_1 -buff 2.0  
-pos align_actives-hits.maegz -neg align_inactives-hits.maegz
```

The `-buff 2.0` option creates a buffer zone of 2.0 Å around the actives, into which the excluded volume spheres are not permitted to penetrate.

The following output should be written to the terminal window:

```
Number of actives = 5  
Number of inactives = 6  
Total number of excluded volume spheres created = 153  
  
Creating excluded volumes file BuildQsarResults/DHRRR_1.xvol  
  
create_xvolClash successfully completed
```

You may import the `DHRRR_1` hypothesis into the Edit Hypothesis panel in Maestro to view the excluded volumes.

You may also import the active and inactive hit files as well, to verify that excluded volume violations would occur only for the inactives.

5.14.3 Creating Excluded Volumes from a Receptor

In this exercise you will create excluded volumes from the atoms in a receptor structure, using the utility `create_xvolReceptor`. This would normally be done in conjunction with a hypothesis that has been created from a ligand bound to the receptor, using the Edit Hypothesis panel in Maestro.

You must supply a receptor structure, or a portion thereof, in a Maestro file. The receptor structure you provide should *not* contain a bound ligand, because the utility considers all atoms in the file when creating excluded volumes.

In the directory `../userFiles`, you should find the following files derived from a co-crystallized complex of p38 MAP kinase:

<code>1M7Q_hypo.def</code>	Hypothesis feature definitions.
<code>1M7Q_hypo.mae</code>	Reference ligand structure (the crystal structure for the p38 inhibitor).
<code>1M7Q_hypo.tab</code>	Primary hypothesis data.
<code>1M7Q_hypo.xyz</code>	Hypothesis site coordinates.
<code>1M7Q_protein.mae</code>	p38 MAP kinase crystal structure (does not contain bound inhibitor).

1. Enter the following command to create the receptor-based excluded volumes:

```
$SCHRODINGER/utilities/create_xvolReceptor  
-hypo ../userFiles/1M7Q_hypo  
-receptor ../userFiles/1M7Q_protein.mae -buff 2 -limit 5
```

In addition to creating a 2 Å buffer zone with the `-buff` option, the thickness of the excluded volume shell is limited by considering p38 atoms that are no more than 5 Å from the reference ligand, with the `-limit` option.

Upon completion, the file `../userFiles/1M7Q_hypo.xvol` is created with a total of 46 excluded volume spheres. You may now import `1M7Q_hypo` into the **Edit Hypothesis** panel in Maestro to view the excluded volumes.

You can import the receptor structure as well to verify that the excluded volume spheres coincide with the positions of the receptor atoms.

Creating and Searching 3D Databases from the Command Line

This chapter demonstrates how 3D databases can be created, modified and searched by running Phase utilities and programs from the command line. It is assumed that you are already familiar with creating and searching 3D databases from the GUI. The exercises in this chapter are therefore focused on running the sequences of scripts, rather than explaining the science. If you are not familiar with Phase or with creating and searching 3D databases from the GUI, we strongly recommend that you work through the tutorial in [Chapter 3](#) so that you are familiar with the terminology that will be used in this tutorial.

In order to run the programs described in this tutorial, Phase 3.1 must be installed on the machine you are currently using, as well as on any remote hosts on which jobs will be run. You must also define the `SCHRODINGER` environment variable so that it points to the directory in which the Schrödinger software is installed.

Some exercises require the creation of a text file. If you type in the data for the file, ensure that you use spaces and not tabs to separate the fields on each line.

Samples of output are given in some of the exercises so that you can check the results. Some of these samples include CPU times, which will not be the same as you see in your output because of differences in hardware and machine load. These times do however give an estimate of the time needed for the operation.

6.1 Database Utilities

3D databases are created and maintained with the following set of utilities, which are stored in `$SCHRODINGER/utilities`. These are:

<code>phasedb_manage</code>	Creates a new 3D database, adds molecules to an existing database, and deletes molecules.
<code>phasedb_confsites</code>	Creates conformers and pharmacophore sites for molecules stored in a 3D database.
<code>phasedb_findmatches</code>	Does setup and cleanup of database searching jobs.
<code>phasedb_subset</code>	Creates and manipulates database subset files.
<code>phasedb_convert</code>	Converts Phase 1.0 databases to Phase 2.0 format, and interconverts storage mode of Phase 2.0 databases.

In addition, the top-level program `phase_dbsearch` should be present in the `$SCHRODINGER` directory. Together, these programs comprise the command-line interface for Phase database creation and searching.

Before running any command line utility, it is a good idea to display the usage message. You can do this for the Phase utilities by entering the name of the utility. Please examine the usage messages for the above utilities so that you are aware of the available options and parameters.

In the exercises below, some of the commands are too long for the text line. The commands are continued on the next line, and indented. When you enter the command, you should type the command on a single line, or if you want to type it on multiple lines, add a backslash `\` to the end of each line but the last.

6.2 Creating a New 3D Database

In this exercise you will create a new 3D database. First, you will create a directory and copy and extract the tutorial file.

1. Change to the directory that you want to store the database.

```
cd mydir
```

This directory should be located on a file system that is accessible to all hosts that need access to the database.

2. Copy the file `$SCHRODINGER/phase-vversion/tutorial/db_tutorial.tar.gz` to this directory

```
cp $SCHRODINGER/phase-vversion/tutorial/db_tutorial.tar.gz .
```

3. Extract the file as follows:

```
gunzip -c db_tutorial.tar | tar xf -
```

4. Change to the `db_tutorial` directory and create a subdirectory to hold the database:

```
cd db_tutorial
mkdir myFirstDB
```

`myFirstDB` will be the root directory for the database, and you will need to specify the absolute path to that directory when running database utilities. For convenience, we shall define an environment variable `TPATH` to hold the absolute path to the tutorial directory.

5. Enter the version of the following command for the shell you are using:

```
csh/tcsh:      setenv TPATH `pwd`
```

```
bash/ksh:     export TPATH=`pwd`
```

6. Verify that TPATH has been assigned correctly:

```
echo $TPATH
```

This command should print the absolute path to the current directory. If you require more than one terminal session to complete the tutorial, or if you switch shells or windows, you must redefine TPATH.

Next you will create a new database named `stdDB` in the directory `myFirstDB`, and populate it with 100 single-conformer molecules, which are contained in the file `db_tutorial/userFiles/singleConfs.mae`.

7. Enter the following command to create a database:

```
$SCHRODINGER/utilities/phasedb_manage -db $TPATH/myFirstDB/stdDB  
-new -mae $TPATH/userFiles/singleConfs.mae.gz -confs false
```

The option `-new` indicates that a new database is being created. The structures are stored in HDF5 format.

The option `-mae` specifies a file that contains structures in Maestro format to add to the database. This file is a gzipped file: you can specify either a plain file or a gzipped file.

The option `-confs false` means that the structure file contains only one conformer per molecule.

This job runs in the foreground because the `-JOB jobName` option was not used. Running `phasedb_manage` as a job is covered in the next section. This operation requires only a few seconds to finish, during which time information about the addition of each molecule is written to the terminal window. The last few lines of this output should be:

```
LIGAND_NAME = block_1/mol_98 # Title = "852130" Confs = 1  
LIGAND_NAME = block_1/mol_99 # Title = "862412_2" Confs = 1  
LIGAND_NAME = block_1/mol_100 # Title = "862427" Confs = 1
```

```
Doing final database commit
```

```
A total of 100 molecules added to database  
Total number of molecules stored = 100
```

```
CPU time = 1.84 sec  
phasedb_manage successfully completed
```

The CPU time will vary depending on your system.

8. Now verify that the appropriate files have been created in the database directory:

```
ls -F $TPATH/myFirstDB
```

You should see the following files and directory:

<code>stdDB_dbInfo.log</code>	Information about changes that have been made to the database.
<code>stdDB_dbversion</code>	Version information for the database (do not modify).
<code>stdDB_feature.ini</code>	Default pharmacophore feature definitions.
<code>stdDB_ligands/</code>	Subdirectory in which all database structure files are stored.
<code>stdDB_master_phase.inp</code>	List of all database records (do not modify).
<code>stdDB_phasedb</code>	SQLite database file that holds auxiliary information.

Each file above is prefixed with the chosen database name, `stdDB`. This naming convention allows you to create more than one database in a given directory, if you so desire. The full unique name for this database is `$TPATH/myFirstDB/stdDB`, and you must supply this name when running the utilities described in the remainder of this chapter.

At this point, you have extracted 100 single-conformer molecules from the file `singleConfs.mae.gz`, and you have stored them in the database in a single HDF5 file:

```
$TPATH/myFirstDB/stdDB_ligands/block_1/block_ct_1.h5
```

Up to 5,000 molecules can be stored in this file. When molecule 5,001 is added, it is stored in the next file, `block_2/block_ct_2.h5` in the directory `stdDB_ligands/`, and so on. You can reduce the number of molecules per block with the option `-blimit maxMol`. This may be useful when running conformer and site generation jobs over multiple processors in a small database.

You could now search this database if you were willing to generate conformers and pharmacophore sites in memory without storing them in the database, a process called *flexible searching*. In the present case, however, you will be storing conformers and sites in the database. Flexible searches are covered in [Section 6.9](#) and [Section 6.10](#).

When you create a database, you are the owner of all the files and directories associated with that database, and you have read/write permissions to the files, and read/write/execute permissions to the directories. You might want other users to have access to the database, or you might want to deny other users access to the database. If you want to be certain that you are the only person who will ever modify a particular database, then you should ensure that write permissions for all other users are removed throughout the database tree. See for more information on setting permissions.

6.3 Adding Molecules to an Existing Database

In this exercise, you will add a set of multi-conformer molecules to the database created in the previous exercise.

The file `$TPATH/userFiles/multiConfs.mae.gz` contains a set of 36 endothelin ligands with pre-generated conformational models. In order to add these structures to the existing database using `phasedb_manage`, you must supply the `-add` option, and you must indicate that the file you are importing contains multiple conformers per molecule by specifying `-confs true`. You will also use the option `-JOB jobName` to indicate that the process should be run as a job.

1. Add molecules to the database with the following command:

```
$SCHRODINGER/utilities/phasedb_manage -db $TPATH/myFirstDB/stdDB
  -add -mae $TPATH/userFiles/multiConfs.mae.gz -confs true
  -JOB add_mol
```

This command runs the job on the machine you are currently logged into, but you could run it on another machine by using the `-HOST host` option, where *host* is an entry name in the file `$SCHRODINGER/schrodinger.hosts`. You cannot split `phasedb_manage` jobs across multiple CPUs because database records must be created serially, and the bulk of the time is used in reading the input structure file.

This job should finish in a few seconds.

2. Examine the end of the file `add_mol_manage.log` to verify that the addition of molecules was successful.

The end of the file should look like this:

```
LIGAND_NAME = block_1/mol_134 # Title = "endo-34"  Confs = 8
LIGAND_NAME = block_1/mol_135 # Title = "endo-35"  Confs = 3
LIGAND_NAME = block_1/mol_136 # Title = "endo-36"  Confs = 3
```

Doing final database commit

```
A total of 36 molecules added to database
Total number of molecules stored = 136
```

```
CPU time = 4.90 sec
phasedb_manage successfully completed
```

You can also examine the file `$TPATH/myFirstDB/stdDB_dbInfo.log` to see the full history of changes made to the database.

In addition, to the “successfully completed” message in the log file, the presence of the file `add_mol_manage.okay` indicates that the job finished successfully. The `.okay` file is now used by various utilities and programs (`phasedb_manage`, `phasedb_confsites`, `phasedb_convert`, `phase_dbsearch`) to confirm the success of a job. This feature may be particularly useful if you plan to write scripts to automate database tasks.

6.4 Creating Conformers and Pharmacophore Sites

So far, you have created a database of 136 molecules, storing a single conformer for each of the first 100, and multiple conformers for the last 36. In this exercise you will create conformers for the first 100 molecules, and create pharmacophore sites for all 136, using the utility `phasedb_confsites`. Conformers are generated using a rapid torsional sampling technique, followed by application of a soft non-bonded potential to eliminate high-energy structures.

The `phasedb_confsites` job is always run as a Schrödinger job, and this job may be split across multiple CPUs using the `-HOST` option. Also, there is a built-in mechanism to restart `phasedb_confsites` if it fails for hardware-related reasons. For more information on this utility, see [Section 13.2](#) of the *Phase User Manual*.

The options `-confs mode` and `-sub dbSubset` provide control and flexibility when creating conformers and pharmacophore sites. You can use the `-confs` option to force conformer creation for all molecules, or only for molecules that are represented by a single conformer. If `-confs` is omitted, conformer generation is skipped and pharmacophore sites are created using the existing structures. The `-sub` option allows you to operate on a subset of molecules in the database (see [Section 6.15](#) for more information on using subsets).

1. Start the `phasedb_confsites` job with the following command:

```
$SCHRODINGER/utilities/phasedb_confsites -confs auto
-JOB auto_confs -db $TPATH/myFirstDB/stdDB
```

You can examine the file `auto_confs_confsites.log` to monitor the progress of the job. Because you are using `-confs auto`, conformers are created only for molecules that are currently stored as single-conformer models. These excerpts from the log file show examples of generation of conformers and reading of conformers:

```
Processing block_1/mol_1 . . .
Number of conformations read = 1
Running in auto mode - Generating conformations using torsional sampling
** Entering Conformation Generator **

Number of rotatable bonds          3
Core rotatable bonds, maxkeep      3          1000
Maximum excitation level           11
Energy cutoff (kcal/mole)          10.000
```

```
Total core conformations          24
Total conformations minimized      21

Number of conformations stored = 21
Updating <TPATH>/myFirstDB/stdDB_ligands/block_1/block_ct_1.h5.tmp
Creating sites for 21 conformations . . .
Updating <TPATH>/myFirstDB/stdDB_ligands/block_1/block_sites_1.h5.tmp
.
.
.
Processing block_1/mol_101 . . .
Number of conformations read = 8
Running in auto mode - Using existing conformations
Creating sites for 8 conformations . . .
Updating <TPATH>/myFirstDB/stdDB_ligands/block_1/block_sites_1.h5.tmp
```

When the job has finished, you should see the following output at the end of the file `auto_confs_confsites.log`:

```
Updating 136 file records . . .
```

```
phasedb_confsites successfully completed
```

2. Verify that the subdirectory `$TPATH/myFirstDB/stdDB_ligands/block_1/` now contains the HDF5 file `block_sites_1.h5`, which holds the coordinates of all pharmacophore sites for all conformations of each molecule.

```
ls $TPATH/myFirstDB/stdDB_ligands/block_1/
```

Database keys that summarize the 2D and 3D information in the sites files are stored in the SQLite database, and are used to avoid searching molecules that can't possibly provide a match, thus speeding up the searching process.

The 136-molecule database is now complete and ready to be searched. See [Section 6.6](#) through [Section 6.15](#) for details on conducting database searches.

6.5 Deleting Molecules from a Database

There may come a time when you need to delete molecules from a database. In this exercise you will run `phasedb_manage` to delete a specified set of database records.

The file `$TPATH/myFirstDB/stdDB_dbInfo.log` contains a full account of all changes made to the database via `phasedb_manage`, including the unique name of each record created and added to the database. You can use the contents of this file to construct a subset of database records for deletion. Here you will delete the last 10 molecules from the database.

1. Create a file with the appropriate database records as follows:

```
grep LIGAND_NAME $TPATH/myFirstDB/stdDB_dbInfo.log | tail -10 >
recordFile
```

The contents of recordFile should be as follows:

```
LIGAND_NAME = block_1/mol_127 # Title = "endo-27" Confs = 6
LIGAND_NAME = block_1/mol_128 # Title = "endo-28" Confs = 6
LIGAND_NAME = block_1/mol_129 # Title = "endo-29" Confs = 18
LIGAND_NAME = block_1/mol_130 # Title = "endo-30" Confs = 6
LIGAND_NAME = block_1/mol_131 # Title = "endo-31" Confs = 11
LIGAND_NAME = block_1/mol_132 # Title = "endo-32" Confs = 14
LIGAND_NAME = block_1/mol_133 # Title = "endo-33" Confs = 4
LIGAND_NAME = block_1/mol_134 # Title = "endo-34" Confs = 8
LIGAND_NAME = block_1/mol_135 # Title = "endo-35" Confs = 3
LIGAND_NAME = block_1/mol_136 # Title = "endo-36" Confs = 3
```

Everything to the right of a pound sign “#” is treated as a comment and thus is ignored.

2. Delete these records from the database with the following command:

```
$SCHRODINGER/utilities/phasedb_manage -db $TPATH/myFirstDB/stdDB
-delete -records recordFile
```

This operation should require only a fraction of a second, and the end of the output to the terminal window should be:

```
LIGAND_NAME = block_1/mol_134 # Deleted
LIGAND_NAME = block_1/mol_135 # Deleted
LIGAND_NAME = block_1/mol_136 # Deleted

A total of 10 molecules deleted from database
Total number of molecules stored = 126

CPU time = 1.67 sec
phasedb_manage successfully completed
```

The database now contains only 126 molecules.

6.6 Database Searching: Background

This section provides a brief background to searching a database for matches to a hypothesis. If you are already familiar with the way Phase works, you can skip to the next section and start the exercises. For a more detailed explanation, see [Chapter 11](#) and [Chapter 13](#) of the *Phase User Manual*.

When a Phase 3D database is searched, the process is formally divided into two steps: finding and fetching. In the find step, the pharmacophore sites of each conformer of a molecule are

searched for geometric arrangements of site points that match the hypothesis in both feature types and intersite distances. We refer to these conformers as *matches*.

In the fetch step, the conformer associated with each match is retrieved from the database and aligned to the hypothesis. We refer to these conformers as *hits*, and there are various parameters and options to control which hits are fetched from the database, given a set of matches. The fetch step is automatically run after a find step.

First and foremost, hits are fetched in order of decreasing fitness, a quantity that measures how well the site points in the matching conformer align to those of the hypothesis, how well the matching vector features (acceptors, donors, aromatic rings) overlay those of the hypothesis, and how well the 3D chemical structure of the hit superimposes, in an overall sense, with the reference ligand conformation.

Excluded volumes are applied in the fetch step. This normally results in elimination of some hits. If excluded volumes exist for a particular hypothesis, you have the option of applying or ignoring them. While a QSAR model does not affect which hits are obtained, it can be applied or ignored when fetching the hits.

The rationale behind formally separating the finding and fetching steps is that the former is usually much more computationally expensive, and there is no need to repeat it if you wish to change only how hits are ordered or filtered. Thus you can submit database searching jobs that run in the `find+fetch` mode, or the `fetch` mode. In the latter case, the match file from a previous `find+fetch` job is used as the source of matches.

A third mode known as `flex`, or flexible searching, is also supported. In this case, conformations and sites are generated in memory as the database is searched, but they are never written to disk. Flexible searching allows users to create databases that contain only a single 3D conformer per molecule. These databases consume far less disk space than standard, multi-conformer databases, but there is a price to be paid, as flexible searching is about 10 times slower than when conformations and sites are stored in the database. Note that no match file is produced in `flex` mode, so it is not possible to run a `flex` job followed by a `fetch` job. See Sections 1.12 and 1.13 for more details on flexible databases.

Two modes, `find+fetch+flex` and `fetch+flex`, allow refinement of matches from pre-computed conformers, through generation of additional conformers in memory using the top-ranked match (i.e., highest fitness) as the seed structure. This nearly always results in identification of matches with higher fitness scores, and it can provide additional hits that satisfy excluded volumes and other filters, even if none of the hits from pre-computed conformers pass the filters.

6.7 Running a find+fetch Database Search

In this exercise you will set up and run a normal database searching job. The `find+fetch` job scans pre-generated pharmacophore sites for geometric matches to a hypothesis, fetches the associated pre-generated conformers from the database, aligns them to the hypothesis, then writes them in Maestro format to a hit file. You will be searching the database created in the preceding sections, so you need to complete those sections if you have not already done so.

1. Change to the directory containing the hypothesis files:

```
cd $TPATH/userFiles
```

This directory contains the following hypothesis-related files, which were created from a set of endothelin ligands:

DHRR_9.def	Pharmacophore feature definitions used to develop the hypothesis.
DHRR_9.mae	Reference ligand conformation, i.e., the full 3D chemical structure of the molecule that gave rise to the hypothesis.
DHRR_9.qsar	QSAR model.
DHRR_9.tab	General information about the hypothesis.
DHRR_9.xvol	Excluded volume definitions.
DHRR_9.xyz	Hypothesis site point coordinates.

The QSAR model file and excluded volumes file are present with any hypothesis only if these characteristics have been defined for the hypothesis. You may examine the contents of all the hypothesis files except for the QSAR model file, as they contain ordinary text, but do not modify these files.

2. Set up the `find+fetch` database searching job with the following command:

```
$SCHRODINGER/utilities/phasedb_findmatches -setup find_fetch  
-db $TPATH/myFirstDB/stdDB -hypo DHRR_9 -mode find+fetch
```

This command creates the main input file `find_fetch_dbsearch.inp`, which contains all the options and parameters for running the `phase_dbsearch` job. For information on these options, see [Section B.13](#) of the *Phase User Manual*. You do not need to change the options for this exercise.

3. Start the database searching job with the following command:

```
$SCHRODINGER/phase_dbsearch -NO_CHECKPOINT find_fetch
```

You can examine the file `find_fetch_dbsearch.log` to monitor the progress of the job. When the job has finished, you should see output like the following in this file:

```
Driver script for parent phase_dbsearch job find_fetch
Current time: Fri Apr 27 21:21:07 2007
Number of CPUs requested = 1
.
.
.
phasedb_match_keys Output: Fri Apr 27 21:21:08 2007
Screening database using 3D keys . . .
Total number of records matched = 56
Creating records file find_fetch_phase.inp
CPU time = 1.03 sec
phasedb_match_keys successfully completed

phase_dbsearch_runseq.pl invoked by job find_fetch to run phase_dbsearch
binary on each block of 1000 database records
.
.
.
Searching database records . . .
Processing mol_1 (1 of 56)
  Number of conformations = 21 (cumulative conformations = 21)
  Number of matches = 28 (cumulative matches = 28)
  Number of hits fetched = 1
Total number of hits stored = 1
  Processing mol_11 (2 of 56)
  Number of conformations = 100 (cumulative conformations = 121)
  Number of matches = 0 (cumulative matches = 28)
  Total number of hits stored = 1
.
.
.
Processing mol_126 (56 of 56)
  Number of conformations = 18 (cumulative conformations = 2346)
  Number of matches = 16 (cumulative matches = 381)
  Number of hits fetched = 1
  Total number of hits stored = 29

*** End of current block of molecules ***

Total number of molecules processed = 56
Total number of conformations searched = 2346
Total number of matches found = 381
Total number of hits stored = 29
Writing 29 hits to find_fetch-hits.list . . .
Writing 29 hits to find_fetch-hits.mae . . .

CPU time = 6.16 sec
```

```
phase_dbsearch successfully completed
No more records to process for job find_fetch
phase_dbsearch_runseq.pl successfully completed for job find_fetch

Driver script for parent phase_dbsearch job find_fetch finished
Current time: Fri Apr 27 21:21:16 2007
Elapsed time = 00:00:09

phase_dbsearch results for find_fetch are complete
```

The prescreen using 3D keys (phasedb_match_keys) reduced the number of records to search from 126 to 56, and that 29 of those 56 records produced hits. So while the prescreen is not 100% effective at identifying the minimum set of records to search, it does eliminate the majority of records that cannot possibly match the hypothesis (in the present case 70 out of 97 such records were eliminated). The prescreen is also quite fast because it is nothing more than a SQL query to the SQLite database.

4. Run the cleanup job:

```
$SCHRODINGER/utilities/phasedb_findmatches -cleanup find_fetch
```

This job checks that the output files are all present, and notifies you if any are missing. It is not strictly necessary, and will be omitted in later exercises.

You can view the 29 hits by importing the file `find_fetch-hits.maegz` into Maestro. If you do so, you should see a number of properties that come from the database search:

```
phasedb index
Hit Source
Ligand Name
Conf Index
Num Sites Matched
Matched Ligand Sites
Align Score
Vector Score
Volume Score
Fitness
Pred Activity(1)
Pred Activity(2)
Pred Activity(3)
```

Most of these are self-explanatory. The QSAR model contains three PLS factors, so predicted activities are reported for the 1-factor, 2-factor and 3-factor regressions. The phasedb index property is the database record index, and the Hit Source property lists the full path to the database, which is useful when hit files from different databases are combined.

6.8 Running a fetch Database Search

In this exercise you will repeat the previous database search, but without applying the excluded volume filter. Since excluded volume violations are checked in the fetch step, there is no need to repeat the time-consuming find step, and you can run a fetch-only job, using the matches that are stored in the file `find_fetch-matches.out`. You will also change some of the other options in the input file.

1. Set up the fetch-only job:

```
$SCHRODINGER/utilities/phasedb_findmatches -setup fetch_only
      -db $TPATH/myFirstDB/stdDB -hypo DHRR_9
      -matchFile find_fetch-matches.out -useExclVol false
      -maxHitsPerMol 5 -maxHits 50 -mode fetch
```

This step results in the creation of the input file `fetch_only_dbsearch.inp`. The contents of the file should be as follows:

```
alignCutoff=1.2
alignPenalty=1.2
alignWeight=1
dbPathName=TPATH/myFirstDB/stdDB
hitFile=fetch_only-hits.mae
hitListFile=fetch_only-hits.list
hypoID=DHRR_9
matchFile=find_fetch-matches.out
maxHits=50
maxHitsPerMol=5
runMode=fetch
useExclVol=false
useHardAlignCutoff=false
useQSARModel=true
useVolumeGroups=false
vectorCutoff=-1
vectorWeight=1
volumeCutoff=0
volumeWeight=1
```

Here, *TPATH* represents the actual path. Many of the options that were present in the `find+fetch` job are now missing. What you see here are only the options that need to be specified when running in `fetch` mode. In fact, the job stops with an error message if you include any options that apply only to `find+fetch` jobs. This strict option checking is done so that you do not run searches expecting that certain options will affect the results, when in fact they won't.

2. Submit the fetch job:

```
$SCHRODINGER/phase_dbsearch -NO_CHECKPOINT fetch_only
```

When the job has finished, the log file `fetch_only_dbsearch.log` should contain output from only the fetch step:

```
Driver script for parent phase_dbsearch job fetch_only
Current time: Mon Apr 30 13:43:21 2007
Number of CPUs requested = 1
.
.
.
Searching existing matches . . .
Processing mol_1
  Number of matches = 28 (cumulative matches = 28)
  Number of hits fetched = 5
  Total number of hits stored = 5
Processing mol_28
  Number of matches = 2 (cumulative matches = 30)
  Number of hits fetched = 2
  Total number of hits stored = 7
.
.
.
Processing mol_126
  Number of matches = 16 (cumulative matches = 381)
  Number of hits fetched = 5
  Total number of hits stored = 50

*** End of current block of molecules ***

Total number of matches processed = 381
Total number of hits stored = 50
Writing 50 hits to fetch_only-hits.list . . .
Writing 50 hits to fetch_only-hits.mae . . .

CPU time = 2.44 sec

phase_dbsearch successfully completed
No more records to process for job fetch_only
phase_dbsearch_runseq.pl successfully completed for job fetch_only

Driver script for parent phase_dbsearch job fetch_only finished
Current time: Mon Apr 30 13:43:28 2007
Elapsed time = 00:00:07

phase_dbsearch results for fetch_only are complete
```

From this output it should be apparent that all matches are fetched for each molecule, but no more than five are added to the hit list. Moreover, the total number of hits is capped at 50, as requested.

If you import the file `fetch_only-hits.mae` into Maestro you will see how the hits are grouped by molecule and sorted by fitness within each group.

6.9 Creating a Flexible Database

If disk space is at a premium, you may want to create a database in which only a single conformation is stored for each molecule. The necessary multi-conformer models and pharmacophore sites can be generated when the database is searched, without ever storing them on disk. In this section, we demonstrate how to create a database that will be searched in this manner, a so-called *flexible* database.

1. Change to the `$TPATH` directory and create a new directory to hold the flexible database:

```
cd $TPATH
mkdir mySecondDB
```

2. Create a new database named `flexDB` in the directory `mySecondDB`, and populate it with the 100 single-conformer molecules contained in the file `db_tutorial/userFiles/singleConfs.mae`:

```
$SCHRODINGER/utilities/phasedb_manage -db $TPATH/mySecondDB/flexDB
-new -mae $TPATH/userFiles/singleConfs.mae.gz -confs false
```

This operation requires only a few seconds to finish, during which time information about the addition of each molecule is written to the terminal window. The end of this output should be:

```
LIGAND_NAME = block_1/mol_98 # Title = "852130" Confs = 1
LIGAND_NAME = block_1/mol_99 # Title = "862412_2" Confs = 1
LIGAND_NAME = block_1/mol_100 # Title = "862427" Confs = 1
```

Doing final database commit

```
A total of 100 molecules added to database
Total number of molecules stored = 100
```

```
CPU time = 1.86 sec
phasedb_manage successfully completed
```

3. Now verify that the appropriate files have been created in the database directory:

```
ls -F $TPATH/mySecondDB
```

You should see the following files:

flexDB_dbInfo.log	flexDB_dbversion
flexDB_feature.ini	flexDB_ligands/
flexDB_master_phase.inp	flexDB_phasedb

See [Section 6.2 on page 92](#) for further details about each file and directory. This 100-molecule database is now ready for searching in flex mode.

6.10 Running a flex Database Search

In this section you will set up and run a flexible database search. You will use the database created in [Section 6.9](#), so you must complete that section if you have not already done so.

1. Change to the directory that holds the hypothesis files:

```
cd $TPATH/userFiles
```

2. Set up the flexible database search job with the following command:

```
$SCHRODINGER/utilities/phasedb_findmatches -setup flex  
-db $TPATH/mySecondDB/flexDB -hypo DHRRR_9 -mode flex
```

This command creates the input file `flex_dbsearch.inp`, whose contents should be as follows:

```
alignCutoff=1.2  
alignPenalty=1.2  
alignWeight=1  
dbPathName=TPATH/mySecondDB/flexDB  
deltaDist=2  
flexAmideOption=vary  
flexMaxConfs=100  
flexMaxRelEnergy=41.8  
flexSearchMethod=rapid  
hitFile=flex-hits.mae  
hitListFile=flex-hits.list  
hypoID=DHRRR_9  
maxHits=1000  
maxHitsPerMol=1  
minSites=5  
preferBigMatches=true  
runMode=flex  
timeLimit=-1  
useDeltaHypo=false  
useExclVol=true  
useFeatureCutoffs=false
```

```
useFeatureRules=false
useHardAlignCutoff=false
useQSARModel=true
useRefLigand=true
useSiteMask=false
useVolumeGroups=false
vectorCutoff=-1
vectorWeight=1
volumeCutoff=0
volumeWeight=1
```

Here, *TPATH* represents the actual path. The options are very nearly the same as for the find+fetch job except that matchFile, and saveMatchFile are absent, and there are four new flex-specific options:

flexSearchMethod=rapid	Indicates whether peripheral rotatable bonds should be sampled one at a time (rapid) or simultaneously (thorough). The default is rapid.
flexMaxConfs=100	The maximum number of conformers to generate. The default is 100.
flexMaxRelEnergy=41.84	The relative conformational energy window in kJ/mol. The default is 41.84 kJ/mol (i.e., 10 kcal/mol).
flexAmideOption=vary	Amide torsion treatment. The default is vary.

Conformers are generated with the same method that is used in phasedb_confsites, so it is possible to identify the same set of hits, whether storing conformers in the database or creating them during the search.

You do not need to change any of the options.

3. Start the flex database search job:

```
$SCHRODINGER/phase_dbsearch -NO_CHECKPOINT flex
```

When the job finishes, the log file flex_dbsearch.log should contain output like the following:

```
Driver script for parent phase_dbsearch job flex
Current time: Tue May 1 11:11:28 2007
Number of CPUs requested = 1
.
.
.
Searching database records with on-the-fly conformer generation . . .
Processing mol_1 (1 of 100)
  Generating conformations . . .
  ** Entering Conformation Generator **
```

```
Number of rotatable bonds           3
Core rotatable bonds, maxkeep       3      1000
Maximum excitation level           11
Energy cutoff (kcal/mole)         9.990
Total core conformations           24
Total conformations minimized       21

Number of conformations stored = 21 (cumulative conformations = 21)
Searching for matches . . .
Number of matches = 28 (cumulative matches = 28)
Number of hits fetched = 1
Total number of hits stored = 1
.
.
.
Processing mol_100 (100 of 100)
Structure lacks the features required to produce a match - skipping mol_100

*** End of current block of molecules ***

Total number of molecules processed = 100
Total number of conformations searched = 3763
Total number of matches found = 139
Total number of hits stored = 4
Writing 4 hits to flex-hits.list . . .
Writing 4 hits to flex-hits.mae . . .

CPU time = 105.03 sec

phase_dbsearch successfully completed
No more records to process for job flex
phase_dbsearch_runseq.pl successfully completed for job flex

Driver script for parent phase_dbsearch job flex finished
Current time: Tue May 1 11:13:19 2007
Elapsed time = 00:01:51

phase_dbsearch results for flex are complete
```

Since the flexible conformation options were identical to those supplied to phasedb_confsites when the standard database was created, we would expect the same results for the flex and find+fetch searches with respect to the first 100 molecules. You can verify that this is so by examining the files flex_dbsearch.log and find_fetch_dbsearch.log. You should observe identical numbers of matches and hits for mol_1, mol_28, mol_29, and mol_43.

6.11 Using Site Masks

In some situations, you might need to choose how many sites are matched in a hypothesis, or whether individual sites are required to match or to not match. For example, you may know that a ligand cannot bind to a particular receptor unless it contains a positive site and an aromatic ring, and that it cannot bind if it contains a hydrophobic site at a particular location.

To enable this sort of searching you must define a *site mask*, which is a file that encodes for each site in the molecule whether a match is required (1), optional (0), or disallowed (-1). For more information on the site mask file, see [Section B.11](#) of the *Phase User Manual*.

In this exercise you will apply a site mask that requires partial matches to match the donor site and the first aromatic site in the hypothesis that you have been using for the previous exercises. You will also match to a minimum of 4 sites out of the 5 in the hypothesis. Matching to a minimum of 4 sites means that 4-point matches are considered for a molecule if no 5-point matches are found. You will use the first database, which contains the conformers and sites in the database.

1. Ensure that the current directory is the directory that holds the hypothesis files, `$TPATH/userFiles`.
2. Create a site mask file named `DHRR_9.mask` in this directory with a text editor.

The file should contain the following lines. Do not use tabs to separate the items.

```
4 D 1
5 H 0
6 H 0
9 R 1
11 R 0
```

3. Set up the database search job as follows:

```
$SCHRODINGER/utilities/phasedb_findmatches -setup site_mask
      -db $TPATH/myFirstDB/stdDB -hypo DHRR_9 -minSites 4
      -mode find+fetch
```

The contents of the input file `site_mask_dbsearch.inp` should show `minSites` set to 4, and `useSiteMask` set to true.

4. Start the database search job as follows:

```
$SCHRODINGER/phase_dbsearch -NO_CHECKPOINT site_mask
```

5. When the job finishes, examine the file `site_mask_dbsearch.log`

A total of 4303 matches were found, and a total of 96 hits were written to `site_mask-hits.mae`. In the original `find+fetch` job, where partial matching was not used, there were only 381 total matches and 29 hits. So by requiring only 4 sites to match, 39 additional hits were found.

6. To verify that every hit in `site_mask-hits.mae` matches both the donor site and the first aromatic site, enter the following command:

```
grep "D(" site_mask-hits.mae
```

This command should produce the output:

```
"D(5) H(8) H(9) R(10) R(-) "  
"D(5) H(-) H(6) R(10) R(9) "  
"D(4) H(5) H(-) R(10) R(11) "  
.  
.  
.  
"D(7) H(-) H(11) R(15) R(16) "  
"D(8) H(10) H(9) R(14) R(15) "  
"D(7) H(9) H(8) R(11) R(13) "
```

The numbers in parentheses are site indices that indicate how the hit matched the hypothesis. For example, `D(5)` indicates that the fifth site in the hit was mapped to the donor site in the hypothesis. When a site in the hypothesis is not matched by the hit, a hyphen appears, for example `H(-)`. There are no hyphens for the donor site or the first aromatic ring site because the site mask required that both of these sites be matched.

6.12 Using Feature-Matching Rules

There may be occasions on which, for example, you want to allow a hydrophobic site in the hypothesis to match either a hydrophobic or an aromatic site in the database you are searching. You might also want to prevent matching of some other kind of pharmacophore, such as an ionizable site, from matching the hydrophobe if the hydrophobic site is *not* matched. These sorts of conditions may be imposed by defining *feature-matching rules*. These rules are described in more detail in [Section B.12](#) of the *Phase User Manual*.

In this exercise, you will create a file that contains feature-matching rules and run a search using these rules.

1. Ensure that the current directory is the directory that holds the hypothesis files, `$TPATH/userFiles`.

If you continued from the previous exercise this should be the current directory.

2. Create a rules file named `DHRR_9.rules` in this directory with a text editor.

The file should contain the following lines. Do not use tabs to separate the items.

```
4 D
5 HR NP
6 H
9 R
11 RH
```

The rule 5 HR NP means that hydrophobic site 5 is permitted to match either a hydrophobic or an aromatic site, but if it does not match either of these, then it cannot match a negative ionizable or positive ionizable site. Likewise the rule 11 RH means that the aromatic site 11 can also match a hydrophobe. Because vector and non-vector features are both permitted at at least one site, all vector scoring is turned off. The use of 3D keys is also turned off because of the mixed features.

The site mask created in the previous section is also being applied, so the donor site and the first aromatic site must be matched.

3. Set up the database search job as follows:

```
$SCHRODINGER/utilities/phasedb_findmatches -setup feature_rules
      -db $TPATH/myFirstDB/stdDB -hypo DHRR_9 -minSites 4
      -mode find+fetch
```

The input file `feature_rules_dbsearch.inp` should show `useFeatureRules` set to true, as well as `minSites` set to 4 and `useSiteMask` set to true.

4. Start the database search job as follows:

```
$SCHRODINGER/phase_dbsearch -NO_CHECKPOINT feature_rules
```

5. When the job finishes, examine the file `feature_rules_dbsearch.log`.

Compared to the previous site mask search, the total number of matches has increased from 4303 to 8567, and the number of hits has increased from 96 to 115. This is because the use of mixed permitted features leads to a greater number of ways to match the hypothesis to a given molecule.

6. Enter the following command to examine the mappings in the hit file:

```
zcat feature_rules-hits.maegz | grep "D("
```

The output of this command should be as follows:

```
"D(6) H(8) H(9) R(10) R(12) "
"D(5) H(6) H(7) R(10) R(12) "
"D(5) H(8) H(9) R(10) R(-) "
```

```
.  
.   
.   
"D(6) R(14) H(-) R(13) H(7) "  
"D(7) H(-) H(8) R(10) H(9) "  
"D(5) R(15) H(-) R(11) R(12) "
```

Among the 115 hits, there are many instances of the first hydrophobic site in the hypothesis being matched to an aromatic ring, e.g., "D(6) R(14) H(-) R(13) H(7) ", and the second aromatic ring in the hypothesis being matched to a hydrophobic site, e.g., "D(7) H(-) H(8) R(10) H(9) ".

Verifying that the prohibitions have been enforced is not so straightforward, but it is possible to do so by visual inspection of the hits that fail to match the first hydrophobic site.

In order to avoid inadvertent application of feature-matching rules in subsequent exercises, remove `DHHRR_9.rules` before proceeding:

```
rm DHHRR_9.rules
```

6.13 Using Feature-Matching Tolerances

Database matches are found by comparing intersite distances from the hypothesis with those of the database conformers. A single tolerance `deltaDist` is applied to all intersite distances, which means that different types of pharmacophore features are treated equivalently. Because certain types of ligand-receptor interactions are stronger and more specific, it often makes sense to define different tolerances on matching different types of features.

To use feature-specific tolerances, you must create a feature cutoff file, which contains tolerances that are applied to the positions of the sites in each match, after aligning the sites to the hypothesis.

1. Create a feature cutoff file named `DHHRR_9.tol` with a text editor in the `userFiles` directory.

The file should contain the following lines. Do not use tabs to separate the items.

```
A 1.50  
D 1.50  
H 2.00  
N 0.75  
P 0.75  
R 2.00  
X 1.50  
Y 1.50  
Z 1.50
```

This file defines a tolerance in angstroms for every possible feature type (including custom features X, Y, and Z). Strictly speaking, only tolerances for D, H and R need to be defined in order to search with the current hypothesis, but you should get into the habit of considering all possible features because you may want to use a single set of tolerances for all of your database searching. If you accidentally omit a feature type that is contained in your hypothesis, a default tolerance of 1.0 is used.

This mechanism does not allow you to assign different tolerances to different instances of the same feature type. So with the current hypothesis, the same tolerance would be applied to both hydrophobic sites, and the same tolerance would be applied to both aromatic sites. [Section 6.14](#) describes how to define hypothesis-specific tolerances that allow you to overcome this restriction.

2. Set up the job with the following command:

```
$SCHRODINGER/utilities/phasedb_findmatches -setup feature_tol  
-db $TPATH/myFirstDB/stdDB -hypo DHRR_9 -mode find+fetch
```

The `useFeatureCutoff` option is set to `true`. The number of sites is not specified on the command line, so the default of all sites is used. Consequently, partial matching will not be used and the site mask will not be applied.

3. Start the job with the following command:

```
$SCHRODINGER/phase_dbsearch -NO_CHECKPOINT feature_tol
```

4. When the job finishes, examine the file `feature_tol_dbsearch.log`.

A total of 333 matches were found, and a total of 28 hits were written to `feature_tol-hits.mae`. This compares to 381 matches and 29 hits when feature-matching tolerances were not used.

5. To ensure that feature-matching tolerances are not used in subsequent exercises, remove `DHRR_9.tol`:

```
rm DHRR_9.tol
```

6.14 Using Hypothesis-Specific Matching Tolerances

When using feature-matching tolerances as described in [Section 6.13](#), it is not possible to distinguish between different instances of the same feature type. So if your hypothesis contains two hydrophobic groups, for example, both would be matched with the same tolerance. There may be cases where you will want to use different tolerances. You can apply hypothesis-specific tolerances by creating a file named `hypoID.dxyz`.

In this exercise you will run a match job using different tolerances for given features.

1. Create a feature cutoff file named `DHRRR_9.dxyz` with a text editor in the `userFiles` directory.

The file should contain the following lines. Do not use tabs to separate the items.

```
4 D 1.50
5 H 2.00
6 H 1.50
9 R 2.00
11 R 1.50
```

The cutoffs in this file place tighter tolerances on matching the second hydrophobic site and the second aromatic site. If you want to know precisely how the hypothesis maps to the reference ligand, you can do so from the Edit Hypothesis panel in Maestro.

2. Set up the job with the following command:

```
$SCHRODINGER/utilities/phasedb_findmatches -setup delta_hypo
      -db $TPATH/myFirstDB/stdDB -hypo DHRRR_9 -mode find+fetch
```

The `useDeltaHypo` keyword is now set to `true`.

3. Start the job with the following command:

```
$SCHRODINGER/phase_dbsearch -NO_CHECKPOINT delta_hypo
```

4. When the job finishes, examine the file `delta_hypo_dbsearch.log`

A total of 328 matches were found and a total of 28 hits were written to `delta_hypo-hits.mae`. The number of matches is reduced by 5 compared to when feature matching tolerances were used, reflecting the fact that we required the second hydrophobic site and second aromatic site to match with a tighter tolerance.

5. To ensure that hypothesis-specific matching tolerances are not used in subsequent exercises, remove `DHRRR_9.dxyz`:

```
rm DHRRR_9.dxyz
```

6.15 Working with Database Subsets

There may be instances when you want to perform operations on only a subset of a Phase database. For example, suppose you wanted to restrict a database search to the set of molecules that produced hits in a previous search. Or perhaps you have added new molecules to a database that already contains conformers and sites, and you wish to run `phasedb_confsites` on only the new molecules. Or maybe you want to create a database subset by performing a logical operation (AND, OR, NOT) on two existing subsets. This section contains exercises on how to

accomplish these sorts of tasks with the aid of the utility `phasedb_subset`. For more information on this utility, see [Section 13.4](#) of the *Phase User Manual*.

You can create subsets from hit files, from logical combinations of existing subsets, and from conformer and site queries. The exercises in the next three subsections demonstrate these capabilities.

6.15.1 Restricting a Database Search to a Subset of Hits

In the first exercise, you will create a subset from a hit file and then restrict a database search to that subset. In this instance, you will be repeating the original search but with a tighter distance matching tolerance. There is no need to search the entire database, because molecules that did not match the hypothesis in the initial search cannot possibly match it if the tolerance is reduced. Therefore, you can restrict the search to the 29 molecules in the original match file.

1. Ensure that the `userFiles` directory is the current directory, and change to it if necessary.

```
cd $TPATH/userFiles
```

This directory should contain the file `find_fetch-hits.maegz`.

2. Create a subset from this hit file with the following command:

```
$SCHRODINGER/utilities/phasedb_subset -db $TPATH/myFirstDB/stdDB  
-hits find_fetch-hits.maegz -out myHits
```

The current directory should now contain the subset file `myHits_phase.inp`, which holds `LIGAND_NAME` records for the 29 molecules in `find_fetch-hits.maegz`. The contents of this file should be as follows:

```
LIGAND_DIR = stdDB_ligands  
LIGAND_NAME = block_1/mol_1  
LIGAND_NAME = block_1/mol_28  
.  
.  
.  
LIGAND_NAME = block_1/mol_125  
LIGAND_NAME = block_1/mol_126
```

Phase subset files always have a name of the form `subset_phase.inp`, and they must contain a `LIGAND_DIR` record in addition to the `LIGAND_NAME` records.

3. Set up the database search job with the following command:

```
$SCHRODINGER/utilities/phasedb_findmatches -setup tight_match \  
-db $TPATH/myFirstDB/stdDB -hypo DHRR_9 -mode find+fetch \  
-sub myHits -deltaDist 1.25
```

In the input file `tight_match_dbsearch.inp`, `deltaDist` is now equal to 1.25, and `dbSubsetFile` is `myHits_phase.inp`.

4. Start the database search job with the following command:

```
$SCHRODINGER/phase_dbsearch -NO_CHECKPOINT tight_match
```

5. When the job has finished, examine the file `tight_match_dbsearch.log`.

Only 21 of the 29 molecules satisfied the stricter distance matching tolerance.

6.15.2 Creating a Subset from Other Subsets with Logical Operations

In this exercise, you will determine which of the 29 molecules in the previous exercise *failed* to satisfy the new tolerance, by performing a logical operation on two subsets.

1. Create a subset from the 21-molecule hit file as follows:

```
$SCHRODINGER/utilities/phasedb_subset -db $TPATH/myFirstDB/stdDB  
-hits tight_match-hits.maegz -out tightHits
```

2. Now use `phasedb_subset` and its logical NOT operator to subtract the 21-member subset from the original 29-member subset:

```
$SCHRODINGER/utilities/phasedb_subset -db $TPATH/myFirstDB/stdDB  
-in1 myHits -in2 tightHits -logic NOT -out missedHits
```

The file `missedHits_phase.inp` should now contain `LIGAND_NAME` records for only the molecules that failed to satisfy the 1.25 Å tolerance:

```
LIGAND_DIR = stdDB_ligands  
LIGAND_NAME = block_1/mol_1  
LIGAND_NAME = block_1/mol_28  
LIGAND_NAME = block_1/mol_29  
LIGAND_NAME = block_1/mol_43  
LIGAND_NAME = block_1/mol_118  
LIGAND_NAME = block_1/mol_119  
LIGAND_NAME = block_1/mol_120  
LIGAND_NAME = block_1/mol_121
```

6.15.3 Restricting Conformer and Site Creation Using Subsets

In this exercise, you will use subsets to restrict conformer and site creation to a portion of a database. You will use the database you created earlier in this chapter.

1. Add the single-conformer molecules in `$TPATH/userFiles/newMol.mae` to the database you created earlier:

```
$SCHRODINGER/utilities/phasedb_manage -db $TPATH/myFirstDB/stdDB  
-add -mae $TPATH/userFiles/newMol.mae -confs false
```

This operation requires only a few seconds to finish. The end of the output to the terminal window should be:

```
LIGAND_NAME = block_1/mol_149 # Title = "serot-23" Confs = 1  
LIGAND_NAME = block_1/mol_150 # Title = "serot-24" Confs = 1  
LIGAND_NAME = block_1/mol_151 # Title = "serot-25" Confs = 1
```

Doing final database commit

A total of 25 molecules added to database
Total number of molecules stored = 151

CPU time = 1.07 sec
phasedb_manage successfully completed

At this point, the database contains 126 molecules with multiple conformers and pharmacophore sites, and 25 molecules with neither. To bring this database up-to-date, we need to rerun `phasedb_confsites`, but we only want to create conformers and sites for the last 25 molecules.

2. Create a subset of molecules that do not yet have pharmacophore sites:

```
$SCHRODINGER/utilities/phasedb_subset -db $TPATH/myFirstDB/stdDB \  
-sites false -out noSites
```

This command creates the subset file `noSites_phase.inp`, with `LIGAND_NAME` records for only the last 25 molecules:

```
LIGAND_DIR = stdDB_ligands  
LIGAND_NAME = block_1/mol_127  
LIGAND_NAME = block_1/mol_128  
.  
.  
.  
LIGAND_NAME = block_1/mol_150  
LIGAND_NAME = block_1/mol_151
```

For this particular database, we could have used the query `-confs false` to achieve the same result. However, some databases contain rigid molecules that are only ever represented by a single conformer, so the two types of queries are not always equivalent.

3. Start the job to create conformations and sites for only the newly added molecules:

```
$SCHRODINGER/utilities/phasedb_confsites -confs all -sub noSites  
-JOB new_mol -db $TPATH/myFirstDB/stdDB
```

When the job finishes, you should see the following output at the end of the file `new_mol_confsites.log`:

```
Updating 25 file records . . .
Updating 25 ligand records . . .
Updating 25 key records . . .

CPU time = 0.86 sec

phasedb_confsites successfully completed

All subjobs successfully completed

Driver script for parent phasedb_confsites job new_mol finished
Current time: Wed May 2 14:20:36 2007
Elapsed time = 00:00:32
phasedb_confsites results for new_mol are complete
```

All 151 molecules in the database now have multiple conformers and pharmacophore sites.

6.16 Working with Database Properties

The Maestro and SD files that you import to create a Phase database usually contain any number of properties for each molecule, and these properties are stored within the structure files in the Phase database. The utility `phasedb_props` can be used to extract the properties stored in the HDF5 structure files and create a separate SQLite property database that can be searched using SQL queries. The utility has two modes: extract mode and query mode. See [Section 13.6](#) of the *Phase User Manual* for more information on this utility.

In this exercise, properties are extracted from the database `$TPATH/myFirstDB/stdDB`, so you must complete [Section 6.2](#) through [Section 6.5](#) and [Section 6.15](#) to obtain the results presented in this section.

A comma-separated value (CSV) file is created each time you run `phasedb_props`, with all properties for all records being written when you extract, and all properties for the matching records being written when you pose a query. When you query the property database, you can create a subset file from the matching records, so Phase database searches can be filtered by properties stored in the database.

1. Ensure that the `userFiles` directory is the current directory, and change to it if necessary.

```
cd $TPATH/userFiles
```

2. Start the job to extract properties with the following command:

```
$SCHRODINGER/utilities/phasedb_props -extract  
$TPATH/myFirstDB/stdDB -props stdDB_props.sqlite  
-csv stdDB_props.csv
```

The job should require only a second or two to finish. When it finishes, the SQLite property database `stdDB_props.sqlite` and the CSV file `stdDB_props.csv` should be present in the current directory.

You can use a text editor or a spreadsheet program to examine `stdDB_props.csv`, which contains all the stored properties for all 151 records in the Phase database. Some of the properties, such as `mol_id` and `num_confs`, were not present in the original structure files, but they can be useful for querying purposes.

The extracted property values come only from the first conformer stored for each molecule, so you should avoid queries that involve 3D-dependent properties, such as `r_mmod_Potential_Energy-MMFF94s`, because the reported value for a given record is not generally representative of the entire conformational ensemble.

Now that the property database `stdDB_props.sqlite` has been created, you can pose a SQL query to find records that satisfy some property filter.

3. Enter the following command to identify molecules with a value of the property `r_user_Activity` greater than 7.5:

```
$SCHRODINGER/utilities/phasedb_props  
-query 'r_user_Activity > 7.5'  
-props stdDB_props.sqlite -csv actives.csv -sub actives
```

The query must be enclosed in quotes so that it is interpreted as a single argument by the shell. When the job finishes, the files `actives.csv` and `actives_phase.inp` should be present in the current directory.

4. Use a text editor or spreadsheet program to examine `actives.csv` and verify that there are 5 matching records, and `r_user_Activity` is greater than 7.5 in each case.
5. Verify that the subset file `actives_phase.inp` contains the same 5 records.

Getting Help

Schrödinger software is distributed with documentation in PDF format. If the documentation is not installed in `$SCHRODINGER/docs` on a computer that you have access to, you should install it or ask your system administrator to install it.

For help installing and setting up licenses for Schrödinger software and installing documentation, see the *Installation Guide*. For information on running jobs, see the *Job Control Guide*.

Maestro has automatic, context-sensitive help (Auto-Help and Balloon Help, or tooltips), and an online help system. To get help, follow the steps below.

- Check the Auto-Help text box, which is located at the foot of the main window. If help is available for the task you are performing, it is automatically displayed there. Auto-Help contains a single line of information. For more detailed information, use the online help.
- If you want information about a GUI element, such as a button or option, there may be Balloon Help for the item. Pause the cursor over the element. If the Balloon Help does not appear, check that Show Balloon Help is selected in the Maestro menu of the main window. If there is Balloon Help for the element, it appears within a few seconds.
- For information about a panel or the tab that is displayed in a panel, click the Help button in the panel, or press F1. The help topic is displayed in your browser.
- For other information in the online help, open the default help topic by choosing Online Help from the Help menu on the main menu bar or by pressing CTRL+H. This topic is displayed in your browser. You can navigate to topics in the navigation bar.

The Help menu also provides access to the manuals (including a full text search), the FAQ pages, the New Features pages, and several other topics.

If you do not find the information you need in the Maestro help system, check the following sources:

- *Maestro User Manual*, for detailed information on using Maestro
- *Maestro Command Reference Manual*, for information on Maestro commands
- *Maestro Overview*, for an overview of the main features of Maestro
- *Maestro Tutorial*, for a tutorial introduction to basic Maestro features
- *Phase User Manual*, for detailed information on using Phase
- Phase Frequently Asked Questions pages, at https://www.schrodinger.com/Phase_FAQ.html

- Known Issues pages, available on the [Support Center](#).

The manuals are also available in PDF format from the Schrödinger [Support Center](#). Local copies of the FAQs and Known Issues pages can be viewed by opening the file `Suite_2009_Index.html`, which is in the `docs` directory of the software installation, and following the links to the relevant index pages.

Information on available scripts can be found on the [Script Center](#). Information on available software updates can be obtained by choosing Check for Updates from the Maestro menu.

If you have questions that are not answered from any of the above sources, contact Schrödinger using the information below.

E-mail: help@schrodinger.com
USPS: Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone: (503) 299-1150
Fax: (503) 299-4532
WWW: <http://www.schrodinger.com>
FTP: <ftp://ftp.schrodinger.com>

Generally, e-mail correspondence is best because you can send machine output, if necessary. When sending e-mail messages, please include the following information:

- All relevant user input and machine output
- Phase purchaser (company, research institution, or individual)
- Primary Phase user
- Computer platform type
- Operating system with version number
- Phase version number
- mmshare version number

On UNIX you can obtain the machine and system information listed above by entering the following command at a shell prompt:

```
$SCHRODINGER/utilities/postmortem
```

This command generates a file named `username-host-schrodinger.tar.gz`, which you should send to help@schrodinger.com. If you have a job that failed, enter the following command:

```
$SCHRODINGER/utilities/postmortem jobid
```

where *jobid* is the job ID of the failed job, which you can find in the Monitor panel. This command archives job information as well as the machine and system information, and includes input and output files (but not structure files). If you have sensitive data in the job

launch directory, you should move those files to another location first. The archive is named `jobid-archive.tar.gz`, and should be sent to help@schrodinger.com instead.

If Maestro fails, an error report that contains the relevant information is written to the current working directory. The report is named `maestro_error.txt`, and should be sent to help@schrodinger.com. A message giving the location of this file is written to the terminal window.

More information on the `postmortem` command can be found in [Appendix A](#) of the *Job Control Guide*.

On Windows, machine and system information is stored on your desktop in the file `schrodinger_machid.txt`. If you have installed software versions for more than one release, there will be multiple copies of this file, named `schrodinger_machid-N.txt`, where *N* is a number. In this case you should check that you send the correct version of the file (which will usually be the latest version).

If Maestro fails to start, send email to help@schrodinger.com describing the circumstances, and attach the file `maestro_error.txt`. If Maestro fails after startup, attach this file and the file `maestro.EXE.dmp`. These files can be found in the following directory:

```
%USERPROFILE%\Local Settings\Application Data\Schrodinger\appcrash
```

Glossary

Active compound—A compound that shows high affinity for the biological target. Synonymous with the term *ligand*.

Active set—The set of active compounds that is used to develop a pharmacophore model. This set does not necessarily include all active compounds.

Excluded volume—A region of space in a pharmacophore hypothesis that should not be occupied by any atom of an active compound.

Feature—see **Pharmacophore feature**

Hit—A structure in a 3D database that is found to contain an arrangement of site points that can be mapped to the pharmacophore hypothesis. A hit is not necessarily active, but it is presumed to have a greater than average probability of being active if it was retrieved using a valid hypothesis.

Hypothesis—see ***n*-Point pharmacophore hypothesis**

Inactive compound—A compound that shows little or no affinity for the biological target.

Intersite distance—The distance between any two site points in a pharmacophore.

Ligand—see **Active compound**

Negative compound—A compound that is inactive, yet highly similar in structure to one or more known actives. Some compounds are negative because they lack certain key pharmacophore features found in true actives. Other negatives may actually satisfy exactly the same pharmacophore hypotheses as the actives, but possess extraneous structural characteristics that prevent binding.

Pharmacophore feature—A characteristic of chemical structure that may facilitate a noncovalent interaction between a ligand and a biological target. Examples are hydrogen-bond acceptor ("A"), hydrogen-bond donor ("D"), hydrophobe ("H"), positive ionic center ("P"), negative ionic center ("N").

Pharmacophore site—The labeling and location of a particular pharmacophore feature within a molecule. For example, a hydrogen bond acceptor site could simply be a nitrogen atom that carries an available lone pair. A hydrophobic site might be a methyl carbon or the centroid of a phenyl ring. The term *site point* is often used interchangeably with pharmacophore site.

Pharm set—The set of all compounds, active and inactive, used to develop a pharmacophore model.

***n*-Point pharmacophore**—Any 3D arrangement of *n* pharmacophore features.

***n*-Point pharmacophore hypothesis**—A specific 3D arrangement of *n* pharmacophore features, with associated uncertainties in the feature positions. High affinity ligands in their active conformations are expected to contain pharmacophore sites that can be mapped (within the limits of uncertainty) to any valid hypothesis. A given hypothesis may contain features that are associated with a single mode of binding, or it may contain features that are common to two or more modes of binding.

Reference ligand—The ligand that provides the pharmacophore that defines a hypothesis. In pharmacophore model development, this pharmacophore yields the highest multi-ligand alignment score for the active-set ligands. The reference ligand matches the hypothesis exactly, and has a perfect fitness score.

Site point—see **Pharmacophore site**

3D Database—A set of molecules, each of which is represented by one or more 3D conformational models, augmented with a pharmacophore-based representation of the molecules. A 3D database includes feature types and site point coordinates for each conformation.

Variant—The set of feature types in a pharmacophore. For example, the variant AHH indicates a 3-point pharmacophore containing one hydrogen bond acceptor and two hydrophobic sites.

Vector feature—A pharmacophore feature that contains directionality, such as a hydrogen bond acceptor, hydrogen bond donor, or aromatic ring. A vector feature does not necessarily have vector geometry.

Vector geometry—the geometric characteristics of hydrogen bond acceptors and donors. Refers to the direction of lone pairs in a hydrogen-bond acceptor or the direction of the heavy-atom–hydrogen-atom bond in hydrogen-bond donors. Features with vector geometry must be vector features.

120 West 45th Street, 29th Floor
New York, NY 10036

101 SW Main Street, Suite 1300
Portland, OR 97204

8910 University Center Lane, Suite 270
San Diego, CA 92122

Zeppelinstraße 13
81669 München, Germany

Dynamostraße 13
68165 Mannheim, Germany

Quatro House, Frimley Road
Camberley GU16 7ER, United Kingdom

SCHRÖDINGER.